

AD-A159 003

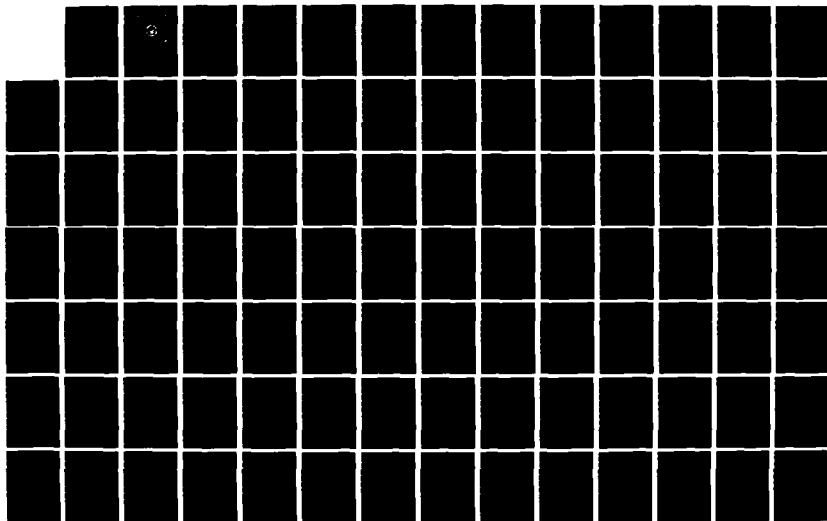
DEVELOPMENT AND ENHANCEMENT OF COMPUTER PERFORMANCE  
MODELING TOOL (CPMT)(U) NAVAL POSTGRADUATE SCHOOL  
MONTEREY CA J G DUARTE JUN 85


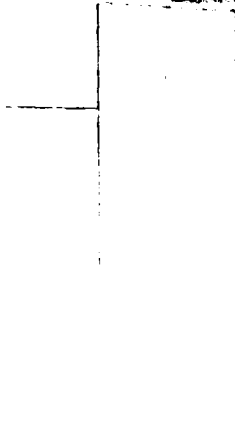
1/2

UNCLASSIFIED

F/G 9/2

NL





2

# NAVAL POSTGRADUATE SCHOOL

Monterey, California



AD-A159 003

DTIC  
ELECT  
SEP 12 1985  
S A

## THESIS

DEVELOPMENT AND ENHANCEMENT  
OF  
COMPUTER PERFORMANCE MODELING TOOL (CPMT)

by

Jose G. Duarte

June 1985

Thesis Advisor:

Alan A. Ross

DTIC FILE COPY

Approved for public release; distribution is unlimited

85 9 10 020

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development and Enhancement of Computer Performance Modeling Tool (CPMT)		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jose G. Duarte		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		12. REPORT DATE June 1985
		13. NUMBER OF PAGES 117
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)  UNCLASSIFIED
		15a. DECLASSIFICATION/DEREGISTRATION SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution is unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  queueing network, simulation model, statistical analysis, validation.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  The Computer Performance Modeling Tool (CPMT) is a queueing network simulator to be used in support of Computer Performance and Evaluation courses like CS4400. This thesis is a continua- tion of the CPMT development project and consists of adaptive and perfective maintenance work to modify the existing simulator to add extended modeling capability and to improve the simulator performance. The thesis effort also included (Continued)		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

1

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT (Continued)

rewriting the CPMT user's manual to reflect new features, establishing a change log for the program and continuing validation of the simulator. *Key words:*

Approved for public release; distribution is unlimited.

Development and Enhancement  
of  
Computer Performance Modeling Tool (CPMT)

by

Jose G. Duarte  
Lieutenant Commander, Portuguese Navy  
B.S., Portuguese Naval Academy, 1970

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL  
June 1985



DTIC COPY INSPECTED 3	
Classification	
Distribution/Availability Codes	
Dist	Avail and/or Special
A1	

Author:

Jose G. Duarte  
Jose G. Duarte

Approved by:

Alan A. Ross  
Alan A. Ross, Thesis Advisor

B. J. Frew  
B. J. Frew, Second Reader

B. J. MacLennan  
B. J. MacLennan, Chairman, ~~421~~  
Department of Computer Science

Kneale T. Marshall  
Kneale T. Marshall,  
Dean of Information and Policy Sciences

## ABSTRACT

The Computer Performance Modeling Tool (CPMT) is a queueing network simulator to be used in support of Computer Performance and Evaluation courses like CS4400. This thesis is a continuation of the CPMT development project and consists of adaptive and perfective maintenance work to modify the existing simulator to add extended modeling capability and to improve the simulator performance. The thesis effort also included rewriting the CPMT user's manual to reflect new features, establishing a change log for the program and continuing validation of the simulator.

## TABLE OF CONTENTS

I.	INTRODUCTION . . . . .	11
A.	BACKGROUND . . . . .	11
	1. Overview of the Computer Performance Evaluation Methods . . . . .	11
	2. Computer Performance Modeling Tool (CPMT) . . . . .	12
B.	OBJECTIVES . . . . .	13
C.	THESIS ORGANIZATION . . . . .	13
II.	MAINTENANCE EFFORT . . . . .	15
A.	TYPE OF MAINTENANCE . . . . .	15
B.	MAINTENANCE PROCESS . . . . .	15
C.	ANALYSIS OF REQUIREMENTS . . . . .	16
	1. Improvement of Processing Efficiency . . . . .	16
	2. Extension of Modeling Capabilities . . . . .	17
	3. Improvement of Run Flexibility . . . . .	20
	4. Enhancement of User Friendliness . . . . .	22
D.	REVIEW OF PROGRAM DESIGN AND IMPLEMENTATION . . . . .	25
E.	TESTING . . . . .	25
F.	UPDATING OF DOCUMENTATION . . . . .	25
III.	CHANGE LOG . . . . .	26
A.	PERFECTIVE CHANGES . . . . .	27
	1. Reduction of Memory Requirements . . . . .	27
	2. Reduction of Processing Time to Produce Statistics . . . . .	30
B.	ADAPTIVE CHANGES . . . . .	32
	1. Capability for Closed Queueing Network Modeling . . . . .	32



2.	Capability for Alternative Queueing Disciplines . . . . .	34
3.	Computation of the Mean Number of Jobs in the System . . . . .	36
4.	Interval for Gathering Statistics . . . . .	39
5.	Computation of the System Throughput . . . . .	42
6.	Alternative Specification of Run Duration . . . . .	43
7.	Capability for Rerunning a Simulation . . . . .	45
8.	Display of Model Specifications . . . . .	46
9.	Printing of Model Specifications . . . . .	47
10.	Updating of Model Specifications . . . . .	48
11.	Deletion and Copy of Simulation Model . . . . .	49
12.	Changing of the Model Specifications . . . . .	50
13.	Handling of Alphabetic Characters . . . . .	51
14.	Printing of Distributions and Queueing Disciplines . . . . .	52
C.	CORRECTIVE CHANGES . . . . .	53
1.	Error in Deleting a Simulation Model . . . . .	53
2.	Error as Executing a Simulation . . . . .	54
3.	Incorrect Handling of Multiservers . . . . .	54
D.	TYPE AND VOLUME OF CHANGES . . . . .	55
E.	IMPACT ON THE CPMT USER'S MANUAL . . . . .	56
IV.	CPMT USER'S MANUAL . . . . .	57
A.	GENERAL DESCRIPTION OF THE CPMT . . . . .	57
B.	MODEL DESIGN AND SPECIFICATION . . . . .	58
1.	Timing . . . . .	59
2.	Servergroup Record . . . . .	59
3.	Job Type Record . . . . .	60
4.	Routing Record . . . . .	61
5.	Distribution Specification . . . . .	62
6.	Routing Design Rules . . . . .	62
7.	Data Input Format . . . . .	64

8.	Example of Model Design . . . . .	64
C.	HOW TO RUN THE SIMULATOR . . . . .	70
1.	Enter a New Simulation Number . . . . .	72
2.	Update Model Specifications . . . . .	72
3.	Check Simulation Specifications . . . . .	73
4.	Run a Simulation Model . . . . .	75
5.	Print Data Base . . . . .	79
6.	Print Data Base for a Single Model . . . . .	79
7.	Delete a Simulation Model . . . . .	79
8.	Copy a Simulation Model . . . . .	80
9.	Display Simulation Model Specifications . . . . .	81
10.	Exit CFMT Environment . . . . .	82
11.	Updating Options . . . . .	83
12.	SUMmary of the Manual Contents . . . . .	94
V.	TESTING AND VALIDATION . . . . .	95
1.	Criteria . . . . .	95
2.	Test Case #1 . . . . .	97
3.	Test Case #2 . . . . .	99
4.	Test Case #3 . . . . .	100
5.	Test Case #4 . . . . .	103
6.	Test Case #5 . . . . .	105
7.	Conclusions . . . . .	113
VI.	CCNCLUSION . . . . .	114
	LIST OF REFERENCES . . . . .	116
	INITIAL DISTRIBUTION LIST . . . . .	117

## LIST OF TABLES

I	Type and Effect of Changes . . . . .	55
II	Critical Values from T-statistical Table . . . . .	96
III	Simulation Results of Test Case #1 . . . . .	98
IV	Mean and Stdv for Test Case #1 . . . . .	98
V	Simulation Results of Test Case #2 . . . . .	99
VI	Characteristics of each Standard Type of Job . .	101
VII	Simulation Results of Test Case #3 . . . . .	102
VIII	Analytical Solution of Test Case #3 . . . . .	102
IX	Mean and Stdv for Test Case #3 . . . . .	103
X	Simulation Results of Test #4 . . . . .	104
XI	Mean and Stdv for Test Case #4 . . . . .	105
XII	Simulation Results of Test Case #5 . . . . .	110
XIII	Mean and Stdv of Test Case #5 . . . . .	112

## LIST OF FIGURES

2.1	Open Queueing Network . . . . .	17
2.2	Closed Queueing Network . . . . .	19
3.1	Mean Number of Jobs in the System . . . . .	38
3.2	Computation of the Accumulated Area . . . . .	39
3.3	Value of the Statistical Flag . . . . .	41
3.4	Start and Stop Areas . . . . .	42
3.5	Correction of the Accumulated Area . . . . .	45
4.1	Queueing Disciplines . . . . .	62
4.2	Distribution Types and Parameters . . . . .	63
4.3	Servergroup Record Data Form . . . . .	64
4.4	Job Type and Routing Record Data Form . . . . .	65
4.5	Computer System H/W Organization . . . . .	66
4.6	Data Parameters . . . . .	67
4.7	Computer System Model . . . . .	68
4.8	Model Servergroup Data Form . . . . .	69
4.9	Model Job Type and Routing Form . . . . .	70
4.10	Master Menu Options . . . . .	71
4.11	Update Menu Options . . . . .	73
4.12	Simulation Specification Error Messages . . . . .	74
4.13	Example of Execute Simulation Model Dialogue . . . . .	76
4.14	Example of Statistical Output Report . . . . .	78
4.15	Example of Display Simulation Model Dialogue . . . . .	82
4.16	Add Job Type Record Dialogue . . . . .	85
4.17	Add Routing Record Dialogue . . . . .	86
4.18	Add Servergroup Record Dialogue . . . . .	88
4.19	Changing Job Type Record Dialogue . . . . .	92
4.20	Change Routing Record Dialogue . . . . .	93
5.1	Branching probabilities Matrix . . . . .	106

5.2	Average Think and Service Times . . . . .	107
5.3	Simulation Model of Test Case #5 . . . . .	108
5.4	Servergroup Data Form for Test Case #5 . . . . .	108
5.5	Job Type and Routing Record Data Form of Test #5 . . . . .	109

## I. INTRODUCTION

The Computer Performance Modeling Tool (CPMT) is a queueing network simulator designed to model computer systems, and it is written in PASCAL for the VAX-11/VMS environment.

This thesis describes a development and enhancement effort to improve the performance and modeling capability of the CPMT simulator.

### A. BACKGROUND

#### 1. Overview of the Computer Performance Evaluation Methods

The application of computer performance and evaluation includes the analysis and enhancement of performance of existing systems and the prediction of performance of planned or projected systems. Performance of existing systems can be evaluated via measurements, using hardware and/or software monitors, either in a user environment or under benchmark conditions. However, the interactions in present day computer systems are so complex that some form of modeling is necessary in order to tune, predict, and understand computer system performance. Performance modeling is also widely used during design and development of new systems.

Networks of queues and Markov chains are the most common representations of computer systems. Queueing models are analyzed by mathematical techniques employing applied probability theory [Ref. 1].

The increased complexity of many computer systems models, as a result of inclusion of different resource

scheduling algorithms, makes the design of models difficult and makes the utilization of mathematical analysis unreasonable. In such cases it is necessary to use a system simulation. A system simulation implies the generation of random inputs and the monitoring of distinct events in the modeled system. Once a model has been formulated, a simulator run tracks the execution of the model as determined by the occurrence of events at discrete time instants. The output of the simulation are random variables. Therefore statistical analysis must be performed to produce a meaningful statement about the validity of the simulation results.

## 2. Computer Performance Modeling Tool (CPMT)

CPMT program development began as a class project for the Computer System Performance Evaluation course, CS 4400, taught at the Naval Postgraduate School.

The objectives were the familiarization of students with simulation program design, and to produce a simulation program which could be used within the class context to model computer systems. The class effort produced the initial program design and two program modules. The CPMT development task was then the topic for a Master's Thesis by LT Karen Pagel [Ref. 2]. The product of this effort was an operational, documented and partially tested simulation program ready to be used as a classroom tool for CS 4400, and as a basis for further development and enhancement.

This thesis is a continuation of the overall CPMT development project, and consists of an adaptive and perfective maintenance effort to modify the existing simulator to add extended modeling capabilities and to enhance the simulator performance.

The thesis effort also included rewriting the user's manual to reflect new features, establishing a change log

for the CPMT program and continuing validation of the simulator.

## **B. OBJECTIVES**

The initial CPMT program was operational and had been used as part of a class exercise for CS 4400. However, from the conclusions listed in the thesis by Lt Pagel [Ref. 2], some weaknesses were detected by program designers and users which justified further program development. From those critics and analysis of other potential improvements, four major types of requirements were identified as desirable areas for enhancement: program efficiency, user friendliness, modeling capability, and simulation run flexibility.

The objective of this thesis was to perform a maintenance effort focused on the areas described above, taking advantage of the readability and modularity of the original CPMT program and its complete documentation.

## **C. THESIS ORGANIZATION**

Chapter 2, Maintenance Effort, describes the maintenance process and is concerned with WHAT and WHY maintenance was performed on the CPMT program. It discusses limitations of the original product and presents the additional requirements and program enhancements to be implemented through the maintenance effort.

Chapter 3, Change Log, is programming oriented and is concerned with HOW the CPMT program was changed and which additional requirements have been implemented, and it includes a description of the design considerations involved and the code affected.

The new version of the CPMT user's manual is provided as Chapter 4 and includes a description and examples of model



design and an explanation of how the program is run. Testing and validation of the CPMT program are discussed in Chapter 5. Hypothetical computer systems studies have been used as test models for validation of the simulator.

The conclusions in Chapter 6 present the current limitations and potential enhancements for continuing the CPMT program development.

## **II. MAINTENANCE EFFORT**

### **A. TYPE OF MAINTENANCE**

Program maintenance is the process by which operational programs are corrected, adapted or upgraded. Adaptive maintenance is performed to modify a program to meet changes or expansion of specifications. Perfective maintenance is performed to enhance performance, processing efficiency or maintainability of operational programs. Most of the maintenance work produced for the CPNT program focused on adaptive and perfective maintenance aspects. Nevertheless, some errors in the original program were discovered and corrected throughout the maintenance process.

### **B. MAINTENANCE PROCESS**

The maintenance process was developed through the following phases:

- analysis of requirements
- review of program design
- translation of new design into code
- testing
- updating of documentation

The work performed during these phases is described in the following sections.

### C. ANALYSIS OF REQUIREMENTS

The purpose of the first step of the maintenance process was to identify and analyze the desirable requirements for the simulation program and to group them according to the maintenance work involved.

The following groups of requirements have been analyzed:

- Improvement of processing efficiency
- Extension of modeling capabilities
- Improvement of simulation run flexibility
- Enhancement of program user friendliness

#### 1. Improvement of Processing Efficiency

One important decision in a simulator design is the computer space and time required to run computer system models. In the original CPMT design, all job and event records which describe the problem to be processed by the simulation are created before starting to process jobs through the simulated system. After all jobs are processed, the program traverses the list of job records to calculate the job statistics. This design decision leads to inefficient use of memory space. Long simulations are not possible on the original program due to large memory space requirements.

In the new version jobs and events are dynamically created as the simulation is being processed and there is only a single event record attached to each job at any time. This record is updated whenever a new event for that job is required. Gathering of job statistics is performed as jobs leave the system, avoiding long lists, and allowing job and event records to be released when they complete.

## 2. Extension of Modeling Capabilities

### a. Closed Queueing Networks

One of the major advantages of simulation is generality. The initial version of the CPMT program can simulate only open queueing network models. These models often are appropriate for communication system modeling and sometimes for computer systems modeling [Ref. 3 :p. 234]. Open networks are characterized by one source of job arrivals and one sink that absorbs jobs departing from the network, as shown in Fig 2.1.

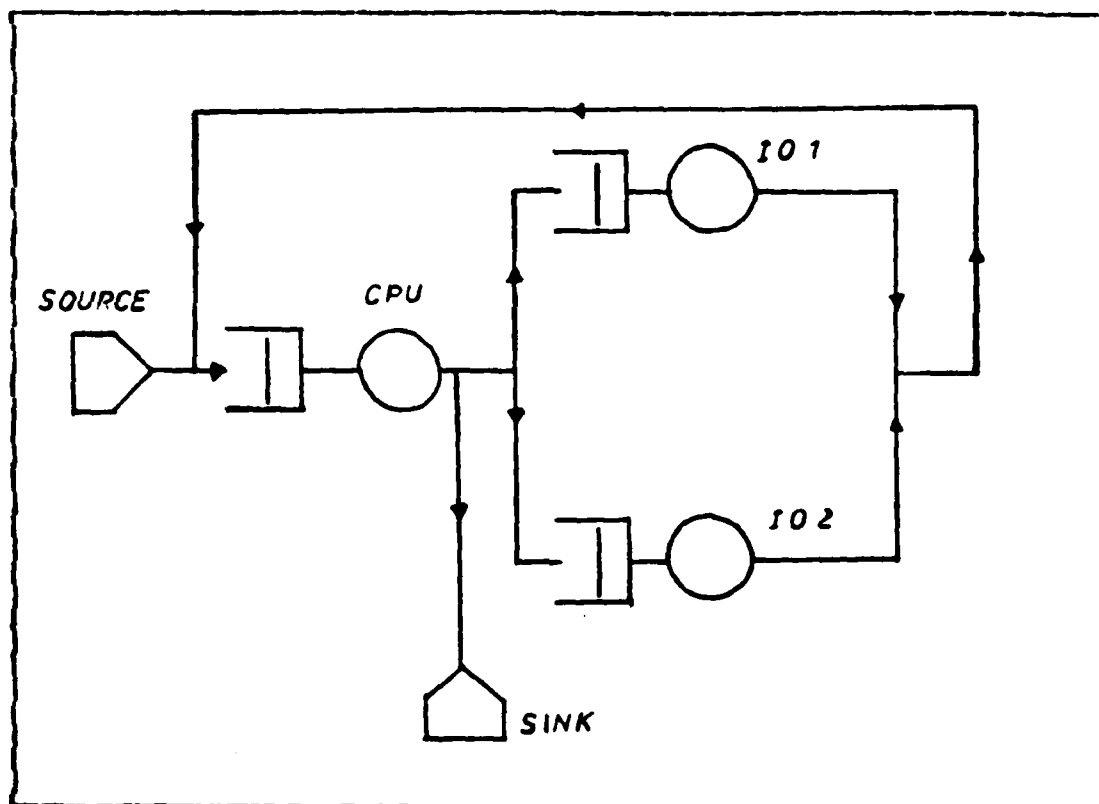


Figure 2.1 Open Queueing Network

One of the implicit assumptions behind these models is that immediately upon its arrival, a job is scheduled into main memory and is able to compete for resources [Ref. 4 :p.423]. In practice the number of main memory partitions in a computer system will be limited which implies the existence of a job scheduler queue. For a large external rate of arrival of jobs, the probability that there is at least one job in this job scheduler queue is very high, and it may be assumed that a job departure immediately triggers the scheduling of an already waiting job into main memory. This situation is often modeled by a closed queueing network, which acts as if the departing jobs wrap around to the input, and immediately re-enter the system. This type of network is shown in Fig 2.2 .

Each circulating job is an active job and must be allocated a specific partition of main memory, and the total number of active jobs is called the degree of multiprogramming. Closed queueing network models have been successfully used to characterize computer systems in a multiprogramming environment [Ref. 3], and can be simulated in the new CPMT Simulator.

#### b. Alternative Queueing Disciplines

A queueing discipline is an algorithm that determines the order in which the jobs in queue for a servergroup of a network are served. A weakness of the initial version of the simulator is that no provision was made for selection of the queueing discipline to be assigned to the servers. Jobs are always served in a first come first served basis. This may not be a good approximation for some computer systems in which other queueing disciplines are implemented in order to improve performance. In the new version of the simulator the following additional queueing disciplines are available to the user, and can be assigned independently to any servergroup.

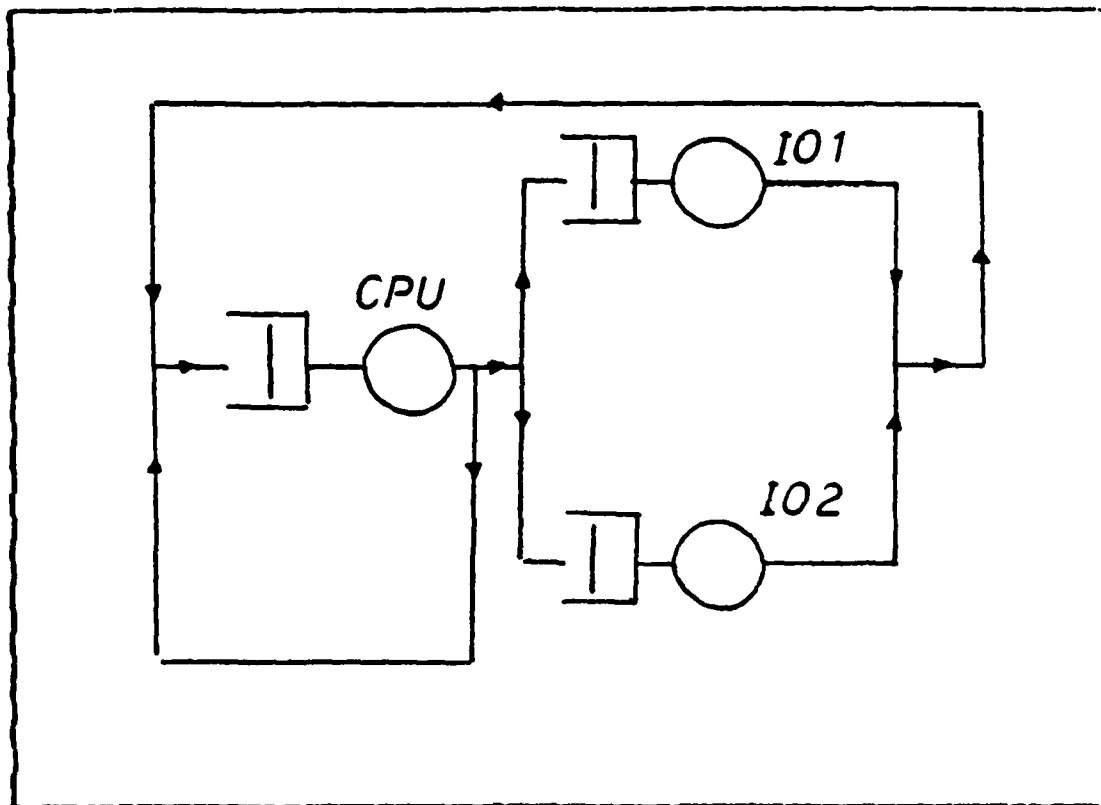


Figure 2.2 Closed Queueing Network

(1) Processor Sharing (PS). All jobs simultaneously receive equal shares of the server. This algorithm is used to model the effect of the round robin queueing discipline with small quantum and overhead times.

(2) Nonpreemptive Priority (NP). Jobs are served in a priority basis but the current service is not interrupted if a higher priority job arrives at the servergroup.

(3) Last Come First Serve (LCFS). Jobs are served in the reverse order of their arrival.

(4) Serve In Random Order (SIRO). Next job to be served is chosen probabilistically, with equal probabilities assigned to all queued jobs.

(5) Short Processing Time First (SPTF). Jobs are served according to the service demand. The smallest service request is served first.

(6) Weighted Short Processing Time First (WSPTF). Jobs are served according to the demand and priority. The job with the smallest request priority ratio is served first.

### c. Measures of Performance

Performance parameters such as system throughput and average number of jobs in the system are not produced by the original simulator.

The system throughput is defined as the number of jobs processed per unit of time. Analysis of the impact of CPU service disciplines, level of programming or number of processors on the system throughput are likely to be performed in the development and design of computer systems.

The mean number of jobs in a queueing system is expressed analytically in terms of probabilities and random variables as described in LAVENBERG [Ref. 1]. For queueing models the mean number of jobs in the system is analytically described by equation 2.1

$$E[n] = \frac{1}{s} \int_{s=0}^s [n_u] du \quad (\text{eqn 2.1})$$

Computation of this value by the simulator is time weighted through the simulation run as described in Chapter 3.

## 3. Improvement of Run Flexibility

### a. Alternative Methods to Specify Simulation Run Duration

One characteristic of simulation programs is that they must provide the timing mechanism for the system

being simulated. A list of coming events is generated and ranked according to time of occurrence. The simulator tracks the list and cycles through the following steps:

- select the event with the next time
- set the clock to this time
- perform action according to the type of occurrence

Simulation run duration can be specified by several methods. The easiest approach consists of specifying the number of times the group of statements which perform the steps described above are to be executed.

Specification of the number of jobs to be processed through the modeled system is an alternative approach and was chosen by the original program designers. This may not be a good solution for closed networks where the number of jobs to be processed is not clearly defined. Another disadvantage of this approach consists of the statistical bias introduced by the shut-down transition phase, when jobs are leaving and none are entering the system. Server utilizations, queue lengths and response time measurements drop in that phase, affecting the final measurements as short simulations are being executed.

The most usual method used to specify run duration consists of defining the total time of the simulation run. One advantage of this approach is to facilitate simulator validation, by allowing comparisons between simulation results and system accounting data gathered for the same period of time.

The new version of CPMT program provides the options: number of jobs, number of events, or simulated time as specification of the simulation run duration for open networks, and the last two alternatives for closed queueing networks.



#### **b. Rerunning a Simulation**

After producing the results of a single simulation run the new version of CPMT will ask whether the user wishes to run the simulation again. If an affirmative response is entered, user will be allowed to enter new values for the simulation run specifications and rerun the model with no need to return to the MASTER MENU.

#### **c. Specification of Period of Time for Statistics**

Statistical bias introduced by simulation execution start-up transition (jobs are entering and none are leaving) and shut-down transition (jobs are leaving and none are entering) is significant when short simulation runs are executed. A statistic oriented feature was implemented in the new CPMT version to reduce or eliminate such effects by providing a special option to the user to specify limits on the interval of time for gathering statistics.

### **4. Enhancement of User Friendliness**

Simulator user interface is a concern of simulation program designers. If it is difficult to interact with a program, the users will be less likely to use it. User friendliness had been implemented in the original simulator, and modifications and additions were accomplished to further improve user program interaction.

#### **a. Display of Model Specifications**

A new option was included in the MAIN MENU to display a specified data base record for a given simulation model, making possible on-line validation of input model specifications.

#### b. Printing of Specifications for a Single Model

An additional option to print the data base for a single model was implemented to improve efficiency and flexibility in accessing data base information.

#### c. Updating of Model Specifications

Different options are now provided to handle user requests depending on whether there is already an user simulation model or not. If a simulation model already exists in the data base, the access to the UPDATE MENU is given from the MASTER MENU. Otherwise, the option to enter the new simulation model number will display the model numbers already existing in data base to prevent collisions and will give direct access to the UPDATE MENU as a new simulation number is entered.

#### d. Copy and Deletion of Simulation Models

Options to copy and delete simulation models were moved from the UPDATE MENU to the MASTER MENU. The scope of the main options in the new version is defined at the simulation model level. Updating of data base records is accomplished by procedures called from the UPDATE MENU rather than the MASTER MENU.

#### e. Changing of Model Specifications

Modification of modeled system specifications is likely to be necessary when using a computer system simulator. No option to change model specifications was implemented in the original CPMT simulation program. In the new version, options to change job type records, routing records and servergroup records are available to the user, as is accessing selected items within records( e.g. job priority within a job type record). Contents of the records

before and after changes are also displayed in order to facilitate record updating.

#### f. Facilities for Exception-handling

A goal of the design of interactive programs is to provide facilities for exception-handling. User errors must be expected, and the user should not be adversely affected by them.

The CPMT program control is driven either by user specification of menu options or user response to prompt messages. The original CPMT version does not accept an alphabetic character as a response for requested options. In such case the program will abort and the user has to restart again. In the new version this will be interpreted as an invalid option, and the menu will be displayed again.

The convention of requiring the uppercase 'Y' for a 'yes' response was also eliminated for the revised program. Both upper and lower case of the letter are assumed to be the affirmative response.

Some input validation is performed by the original program, to insure that input values are within bounds set by either program specification (e.g. menu options) or simulation specification (e.g. mean service time). Additional input validation is accomplished by the new version to prevent abnormal program termination.

#### g. Printing of Specifications

Distribution types and queueing disciplines are printed on the screen and written to output files as meaningful data rather than numerical values, to improve readability of program output.

#### **D. REVIEW OF PROGRAM DESIGN AND IMPLEMENTATION**

For design and implementation of changes, a schedule was established according to the following priority basis:

- Simulator modeling capability
- Program efficiency
- Simulation run flexibility
- Program user friendliness

The baseline for design solutions was to minimize the impact of changes on the program modularity and data structures. The algorithms and implementation considerations are described in Chapter 3.

#### **E. TESTING**

Program testing was performed throughout the change implementation. A few errors were found in the original program and fixed during testing activity. Verification of program correctness under new processing efficiency requirements was performed by comparison of the execution of the new program with the execution of the original program, followed by analysis of results. The quality of the program as a simulation tool is discussed in Chapters 5 and 6.

#### **F. UPDATING OF DOCUMENTATION**

Technical and user documentation was updated to reflect changes in program code and simulator operation. In addition to rewriting the comments in the listing file, a change log was created to facilitate further program maintenance. The change log and the new version of the CPMT user's manual are presented in the next two chapters of this thesis.

### III. CHANGE LOG

In order to provide information necessary to understand the current modifications and trace the evolution of the CPMT program from the initial configuration, a change log was created. This chapter summarizes and presents the contents of the log. Entries to the log include the following information, if applicable :

- Change number
- Type of maintenance (Corrective, adaptive or perfective)
- Type of requirement
- Brief description of requirement or anomaly
- Change design
- Changes in records and data items
- Files affected
- Modules affected
- Procedure and/or Functions eliminated or changed
- New procedures and/or Functions
- Explanation
- Impact on the program modularity, clarity etc.

Changes implemented as a result of this thesis effort are described in the next sections, grouped by type of maintenance and classification of requirements as described in Chapter 2. Change numbers were sequentially assigned for easier reference. Statistics about the type and volume of

maintenance are presented in the last section of this chapter.

#### A. PERFECTIVE CHANGES

##### 1. Reduction of Memory Requirements

###### a. CHANGE #1

###### b. Change Design

Dynamic creation and allocation of job and event records as a simulation is being processed.

###### c. Change Dictionary

Items NEXT\_SERV and COMPLETE were created for the JOB TYPE RECORD, to identify the servergroup at which the next processing event will take place, and detect the job completion; the item FIRST\_JOB\_PART of the same record was eliminated; items NEXT\_JOB\_PART and SCHEDULED were eliminated from the EVENT RECORD.

###### d. Files Affected

CENT.PAS

CJS.PAS

EXT.PAS

###### e. Modules Affected

CREATE JOB STREAM

EXECUTE AND TABULATE

###### f. Procedure Eliminated

CREATE\_JOB\_STREAM

**g. Procedures Changed**

CREATE\_JOB  
EXECUTE\_AND\_TABULATE  
DEPART\_FROM\_SG  
JCE\_ARRIVAL  
JOB\_COMPLETED  
ARRIVE\_AT\_SG

**h. New Procedures**

FIND\_JOB\_TYPE  
CREATE\_INITIAL\_JOBS  
CREATE\_NEW\_EVENT  
EXECUTE

**i. Explanation**

There is no job stream in the new design, thus the procedure CREATE\_JOB\_STREAM was eliminated. The name of the respective module was not modified for easier reference. The new input for the EXECUTE\_AND\_TABULATE module is the linked list of job records created by the new procedure CREATE\_INITIAL\_JOBS, and consists of one job of each job type of the simulated model. Each job record is created by the modified procedure CREATE\_JOB, and has only one associated EVENT record which stores the information required for the first event of that job. Creation of events during a simulation run is requested from the procedure EXECUTE\_AND\_TABULATE as a job departs from a servergroup.

Creation of jobs during a simulation run depends on the type of network being simulated. For the original program capability, open networks, the process is as follows: as a job arrives to the servergroup 0 (dummy server), the procedure JOB\_ARRIVAL invokes first the new procedure FIND\_JOB\_TYPE in order to access in the data base

the JCB TYPE record with the same job type, and then calls the procedure CREATE\_JOB to create a new job. Allocation of job and event records for the new job depends on whether there are records available or not. As a job completes, the job type record is referenced by a pointer, and a flag is set to notify the procedure CREATE\_JOB that there is no need to allocate new records. In this case, the CREATE\_JOB procedure updates the job and event records for the new job. Otherwise, new job and event records will be allocated. The arrival time of the new job is computed as the arrival time of the arrived job plus the interarrival time calculated in the procedure CREATE\_JOB. The job record is attached to the arrival queue for the servergroup 0 and becomes ready to be processed. A counter is incremented to keep track of the number of jobs processed.

As referenced above, there is a single event record associated with each job record. That record is updated by the new procedure CREATE\_NEW\_EVENT which is called by the procedure DEPART\_FROM\_SG. As a job departs from a servergroup, the number of the next servergroup to which the job is routed is checked. If that servergroup is not the exit servergroup (SG10) a new event is created for that job.

A new procedure EXECUTE was created within the procedure EXECUTE\_AND\_TABULATE for easier program readability. This procedure handles the processing loop of job departures and arrivals and calls the procedures DEPART\_FROM\_SG and ARRIVE\_AT\_SG.

#### j. Impact on the Program Modularity

Program modularity was affected by this change. Procedures CREATE\_JCB, CREATE\_NEWEVENT and FIND\_JOB\_TYPE from the module CREATE JOB STREAM are called by procedures from the EXECUTE\_AND\_TABULATE module.



## 2. Reduction of Processing Time to Produce Statistics

### a. CHANGE #2

### b. Change Design

There is no longer a need to traverse a linked list of job records for gathering statistics; information is collected as jobs complete. Standard deviation computations in the new design are calculated by a different algorithm that is described in the change explanation.

### c. File Affected

EXT.PAS

### d. Module Affected

EXECUTE AND TABULATE

### e. Procedures Eliminated

STD\_DEV

STD\_DEV\_JOB\_TYPES

### f. Procedures Changed

JCE\_COMPLETED

STATS\_FOR\_JOBS

STATS\_FOR\_JOB\_TYPES

### g. New Procedure

INITIALIZE\_STATS

### h. Explanation

A new procedure INITIALIZE\_STATS was created to initialize the statistical counters and accumulator variables.

As each job completes, the values of the statistical counters and accumulator variables are updated; as the processing of jobs is completed, procedures STATS\_FOR\_JOBS and STATS\_FOR\_JOB\_TYPES compute and print statistics for all the jobs and for each job type.

For computation of standard deviations let T be defined by equation 3.1 :

$$T = \sum_{i=1}^N (X_i - \text{MEAN})^2 \quad (\text{eqn 3.1})$$

Using binomial theorem in equation 3.1 , T can be expressed as:

$$T = \sum_{i=1}^N X_i^2 - 2 * \text{MEAN} * \sum_{i=1}^N X_i + N * \text{MEAN}^2 \quad (\text{eqn 3.2})$$

$$\text{MEAN} = \sum_{i=1}^N X_i / N \quad (\text{eqn 3.3})$$

Substituting  $\sum_{i=1}^N X_i$  from equation 3.3 and simplifying equation 3.2 , T can be defined as:

$$T = \sum_{i=1}^N X_i^2 - N * \text{MEAN}^2 \quad (\text{eqn 3.4})$$

$$\text{VARIANCE} = \sum_{i=1}^N (X_i - \text{MEAN})^2 / N \quad (\text{eqn 3.5})$$

$$\text{STANDARD DEVIATION} = \sqrt{\text{MEAN}} \quad (\text{eqn 3.6})$$

Using equations 3.4 , 3.5 and 3.6 , STANDARD DEVIATION can be expressed by the following equation :

$$\text{STANDARD DEV} = \sqrt{(\sum_{i=1}^N X_i^2 - N * \text{MEAN}^2) / N} \quad (\text{eqn 3.7})$$

Based on eqs 3.3, 3.4 and 3.7 the following algorithm was implemented:

--- At the  $i$ th job completion compute the square of the current variable  $X$  and update the statistical accumulator  $\sum X_i^2$

--- As the processing of jobs is finished ( $N$ th job completion), compute the values of the MEAN,  $T$  and STANDARD DEVIATION.

## B. ADAPTIVE CHANGES

### 1. Capability for Closed Queueing Network Modeling

#### a. CHANGE #3

#### b. Change Design

Dynamic creation of jobs is accomplished by two distinct methods whether the model being simulated is an open or closed network. For open networks, as described in change #1, the linked list of jobs created by the procedure CREATE\_INITIAL\_JOBS consists of one job for each job type. If a closed network is being simulated the number of jobs for each type created by the CREATE\_INITIAL\_JOBS procedure depends on the model specification and is determined by the level of programming for each job type. Furthermore, for closed network modeling, a job completion will force the arrival of an identical job into the system.

#### c. Files Affected

CJS.PAS

UFMOD.PAS

EXT.PAS

CFMT.PAS

d. Modules Affected

CREATE JOB STREAM  
UPDATE  
EXECUTE AND TABULATE  
MAIN DRIVER

e. Procedures Changed

ADD\_JOB\_TYPE  
EXECUTE\_AND\_TABULATE  
DEPART\_FROM\_SG  
JCE\_ARRIVAL  
JOB\_COMPLETED

f. New Procedure

CHECK\_NET\_TYPE

g. Explanation

The program processes jobs according to the simulation number assigned to the modeled system. A new procedure CHECK\_NET\_TYPE returns the value of a boolean variable determined by the simulation model number used as input. Simulation models 1 through 49 are assigned to open networks and 50 through 100 to closed network modeling.

These ranges can be easily changed by modifying the bound assigned in the procedure CHECK\_NET\_TYPE

For closed networks the arrival time assigned to job records is not dependent on the user specifications, but rather determined by the procedure which drives the creation of the new job. For jobs created by the procedure CREATE\_INITIAL\_JOBS the arrival time is one <sup>1</sup>, otherwise (jobs created by the procedure JOB\_COMPLETE) the arrival

---

<sup>1</sup>The deterministic and short interarrival time was chosen by the program designer to reduce the elapsed time to initialize the closed network

time for the new job will be the time the completed job leaves the system. A flag is set at each job completion to define the instant a new job has to be scheduled. The procedure DEPART\_FROM\_SG will force a new event that will be a departure from the servergroup 0.

## 2. Capability for Alternative Queueing Disciplines

### a. CHANGE #4

### b. Change Design

The queueing discipline to be observed at a given servergroup is specified by the user as he is adding routing records to a job type. Whenever a job arrives to a servergroup that has a waiting queue, it is inserted according to the queueing discipline specified for that servergroup.

### c. Change Dictionary

New items REC\_DISC and Q\_DISC were included respectively in the DATA BASE and SERVER records to identify the queueing discipline assigned to the servergroup.

### d. Files Affected

RECFILE.DAT

EXT.PAS

CENT.PAS

### e. Modules Affected

UPDATE

EXECUTE AND TABULATE

f. Procedures Changed

ADD\_ROUTING\_RECORD  
IC\_EDIT  
PRINT\_R  
BUILD\_LL\_FRCH\_DB  
PROCESS\_ROUTING\_DATA  
EXECUTE\_AND\_TABULATE  
CREATE\_SERVER\_GROUPS  
ARRIVE\_AT\_SG  
DEPART\_FROM\_SG  
INSERT\_IN\_SG\_Q  
ATTACH\_JOB\_TO\_SERVER

g. New Procedures

SG\_Q\_INSERT\_FRONT  
SG\_Q\_INSERT\_PRIORITY  
SG\_Q\_INSERT\_PROC\_TIME  
SG\_Q\_INSERT\_WEIGHT  
SG\_Q\_INSERT\_RANDOM

h. Explanation

The queueing discipline assigned to a servergroup is stored in the database as the user adds routing records to a job type record. As the linked list for routing jobs is created (procedure CREATE\_ROUTING\_RECORD), the values of the queueing disciplines are stored in a one dimensional array. The procedure CREATE\_SERVER\_GROUPS reads from that array the discipline that is to be assigned to each servergroup, and stores it into the respective servergroup record.

The selection of procedures for implementation of the scheduling algorithm to be observed at a servergroup is performed by the procedure INSERT\_IN\_SG.

The procedures to implement the Last Come First Served (LCFS), Nonpreemptive Priority (NP), Lowest Processing Time First (LPTF), and Lowest Weighted Processing Time First (LWPTF) algorithms are self explanatory.

Simulation of random service is accomplished by random insertion of jobs in the waiting queue; the position in which a job is inserted is computed from the function GENERATE\_VALUE, using the number of queued jobs that is stored in the servergroup record.

The PROCESSOR SHARED implementation is based on the algorithm described in SAUER and CHANDY [Ref. 4 :p.200], and it is distributed across the procedures SG\_Q\_INSERT\_PROC\_TIME, ATTACH\_JOB\_TO\_SERVER, and INSERT\_IN\_SG\_Q. The job with the smallest processing time must be the first to leave the queue and so, the smallest processing time first algorithm is used for insertion into the waiting queue. Computation of the service time depends on the number of jobs waiting to be served and is equal to that number times the request time of the current job being served. When the job completes service, that amount of time is subtracted from the request of each job in the queue, if any, to obtain the job remaining requests.

#### i. Impact on the Program Modularity

The procedure INSERT\_IN\_SG\_Q in the EXECUTE AND TABULATE module calls the function GENERATE\_RANDOM\_VALUE outside the module, to generate a random position for insertion into the waiting queue.

### 3. Computation of the Mean Number of Jobs in the System

#### a. CHANGE #5

b. Change Design

The algorithm for a time weighted computation of the number of jobs in the system is described in the change explanation.

c. File Affected

EXT.PAS

d. Module Affected

EXECUTE AND TABULATE

e. Procedures Changed

JCE\_ARRIVAL

JOB\_COMPLETED

STATS\_FOR\_JCBS

STATS\_FOR\_JOB\_TYPES

f. Explanation

As illustrated in Figure 3.1 the value of the mean number of jobs depends on the time accumulated value of the area under the curve,  $A_n$ . The value of  $A_n$  at the instant  $t_i$  is computed from equation 3.8 where  $t_i$  is the time of a job arrival or departure, and  $N_{i-1}$  is the number of jobs in the system at time  $t_{i-1}$ .

The algorithm used for computation of the mean number of jobs in the system was implemented for all jobs and for each individual job type. Computation of the value of  $A_n$  at a given time is performed either by the procedure JOB\_ARRIVAL or JOB\_COMPLETED depending on if the event is an arrival to the system or a job completion. The number of jobs in the system at times  $t_i$  and  $t_{i-1}$  are stored in the array TOTAL\_JOBS\_SYS, and the values of  $t_i$  and  $t_{i-1}$  are stored in the array INTEREVENT. As the value of  $A_i$  is calculated, the



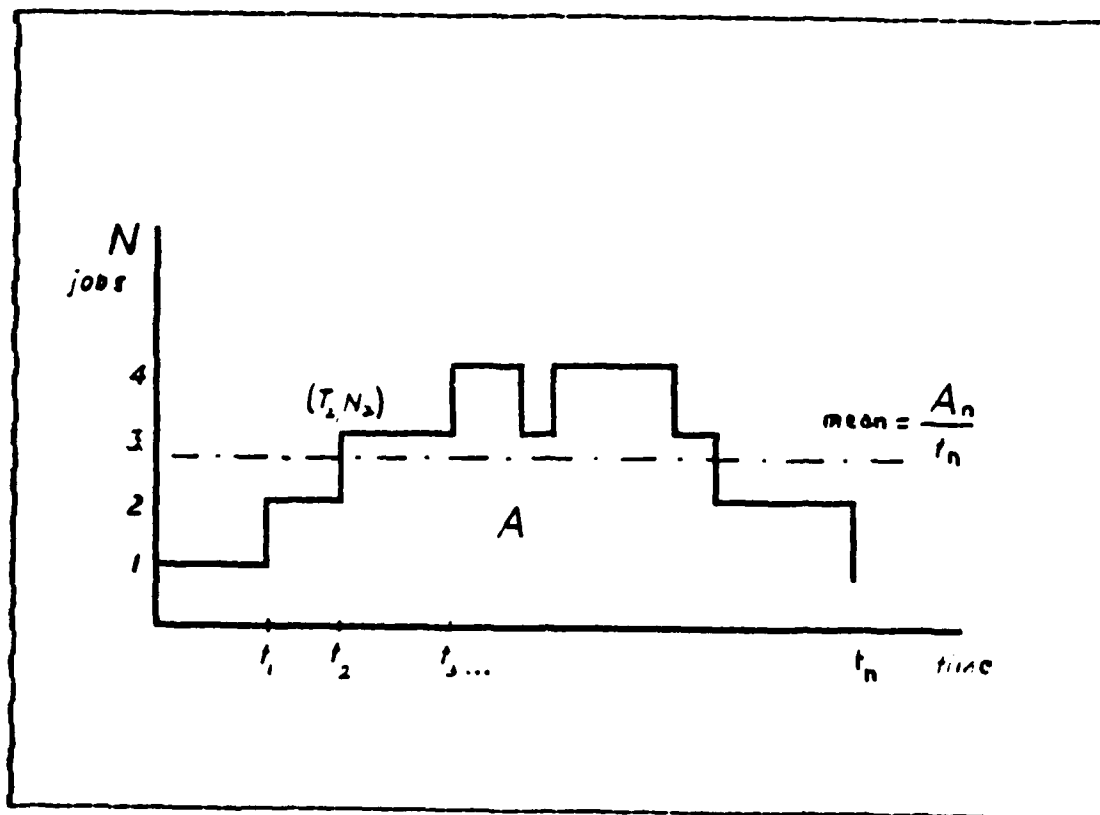


Figure 3.1 Mean Number of Jobs in the System

$$A = \sum_{i=1}^n (t_i - t_{i-1}) * N_i \quad (\text{eqn 3.8})$$

values of  $N_{i-1}$  and  $t_{i-1}$ , which are no longer required, are replaced by the values  $N_i$  and  $t_i$  to prepare for the next computation. The example shown in Figure 3.2 illustrates the application of the algorithm.

The last step, which computes the mean value, consists of dividing the accumulated area by the simulation time. This step is performed by the procedures STATS\_FOR\_JOBS and STATS\_FOR\_JCB\_TYPES.

1) Initial data

t = 50

N = 4

AREA = 12

INTEREVENT(0)	=	50
INTEREVENT(1)	=	**
TOTAL_JOBS_SYS(0)	=	4
TOTAL_JOBS_SYS(1)	=	**

2) At time 55 occurs a job departure

t = 55

N = 3

INTEREVENT(0)	=	50
INTEREVENT(1)	=	55
TOTAL_JOBS_SYS(0)	=	4
TOTAL_JOBS_SYS(1)	=	3

3) Computation of the new AREA at time 55

AREA = AREA +

TOTAL\_JOBS\_SYS(0) \* (INTEREVENT(1) - INTEREVENT(0))

AREA = 12 + 4 \* ( 55 - 50 )

AREA = 32

4) Preparation for the next computation

INTEREVENT(0)	=	55
INTEREVENT(1)	=	**
TOTAL_JOBS_SYS(0)	=	3
TOTAL_JOBS_SYS(1)	=	**

Figure 3.2 Computation of the Accumulated Area

4. Interval for Gathering Statistics

a. CHANGE #6

b. Change Design

Updating of the job and servergroup records as a simulation is being processed is performed over a period of time specified by the user.

c. Files Affected

CIET.PAS  
EXT.PAS  
MESSAGES.DAT

d. Modules Affected

MAIN DRIVER  
EXECUTE AND TABULATE

e. Procedures Changed

STATS\_FOR\_JOBS  
STATS\_FOR\_JOB\_TYPES  
STATS\_FOR\_SERVERGROUPS  
JCE\_ARRIVAL  
JOB\_IN\_SG\_Q  
NO\_JOB\_IN\_SG\_Q  
JOB\_COMPLETED  
ATTACH\_JOB\_TO\_SERVER  
ATTACH\_FIRST\_IN\_Q  
INSERT\_IN\_SG\_Q

f. Explanation

Gathering of information from job and servergroup records to produce statistics is performed depending on whether a flag is set or not. The flag is implemented by the boolean variable STATS and its value depends on the simulation run specifications selected by the user, as shown in the diagram of Figure 3.3 .

As the simulator timing mechanism is driven by events, the specification of the interval for gathering statistics introduces the need of a correction in the computation of the number of jobs in the system, as illustrated in Figure 3.4.

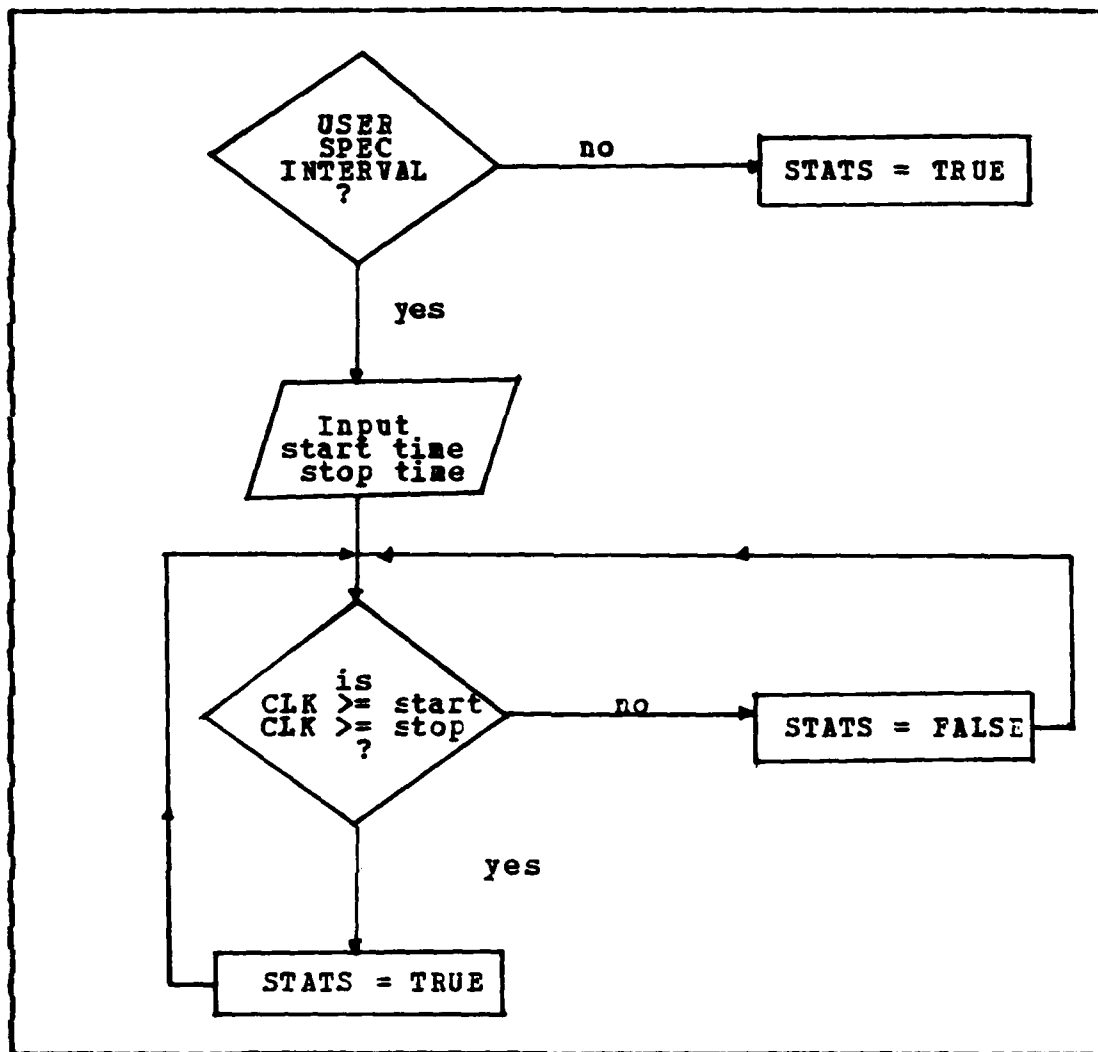


Figure 3.3 Value of the Statistical Flag

As statistics start up and shut down, the areas  $A_1$  and  $A_2$  are calculated at the occurrence of events  $t_b$  and  $t_c$ , using equations 3.9 and 3.10 where  $N_a$  and  $N_c$  are the number of jobs in the system at times  $t_a$  and  $t_c$ . Computation of areas  $A_1$  and  $A_2$  and respective summation to the total accumulated area  $A$  are performed either by the procedure `JOB_ARRIVAL` or `JOB_COMPLETED` depending on whether the events  $t$  and  $t$  are job arrivals to the system or job completions.

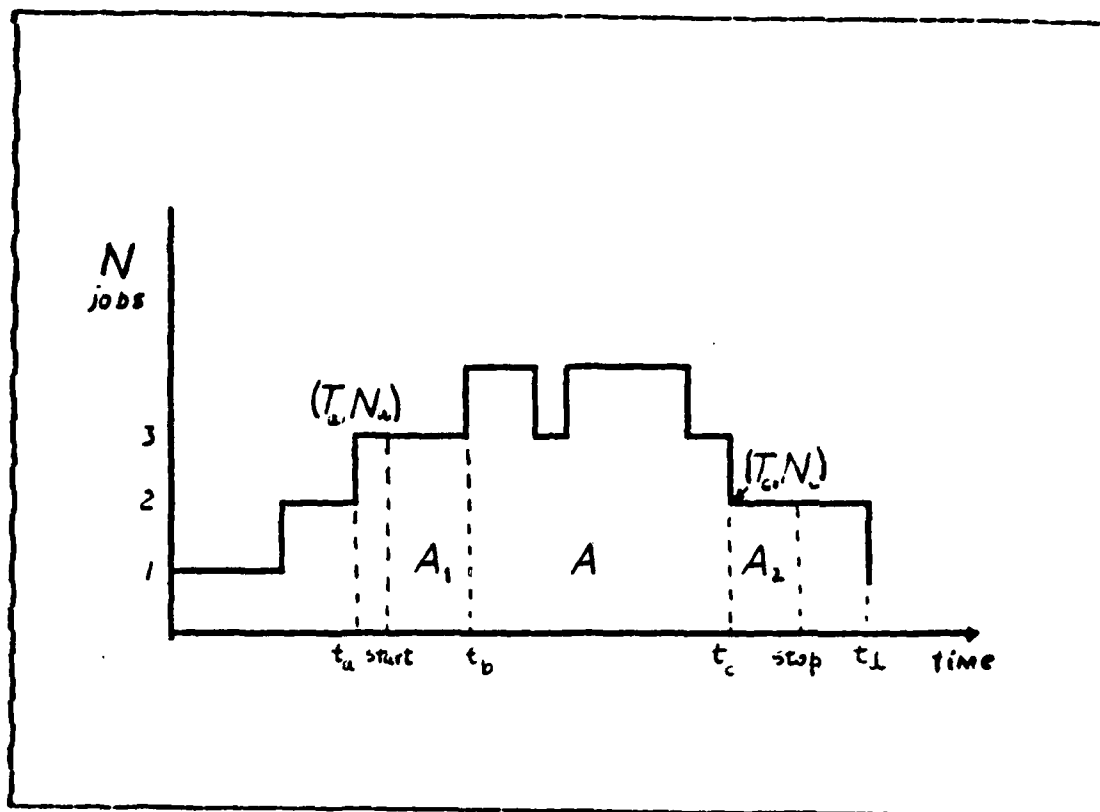


Figure 3.4 Start and Stop Areas

$$A1 = (t_b - \text{start\_stats}) * N_a \quad (\text{eqn 3.9})$$

$$A2 = (\text{stop\_stats} - t_c) * N_c \quad (\text{eqn 3.10})$$

## 5. Computation of the System Throughput

### a. CHANGE #7

### b. Change Design

Accounting of the number of job completions for all jobs and by individual job type; division of those numbers by the elapsed time to determine the rate of throughput.

c. File Affected

EXT.PAS

d. Module Affected

EXECUTE AND TABULATE

e. Procedures Changed

JOB\_COMPLETED

STATS\_FOR\_JOBS

STATS\_FOR\_JOB\_TYPES

f. Explanation

Statistical counters in the procedure JOB\_COMPLETED keep accumulating the number of job completions as the simulation is being processed. The procedures STAT\_FOR\_JOBS and STATS\_FOR\_JOB\_TYPES compute the throughput values, by dividing the total number of completions by the simulated time.

6. Alternative Specification of Run Duration

a. CHANGE #8

b. Change Design

When a simulation run is to be processed, the user specifies the option for run duration. The option is either to end the simulation after a specified number of events or after a specified simulation time. Different conditions are set for controlling the number of iterations of the processing loop depending on the choice.

c. File Affected

EXT.PAS

CIET.PAS

d. Modules Affected

EXECUTE AND TABULATE  
MAIN DRIVER

e. Procedures Changed

EXECUTE\_AND\_TABULATE  
JOE\_ARRIVAL

f. Explanation

In the original program the run duration is always defined by the number of jobs to be processed, and the execution of the main processing loop in the procedure EXECUTE\_AND\_TABULATE terminates when there are no pending events to be processed in the servergroups. The alternative conditions created for controlling the processing loop are enabled by the user specification of either the number of events or simulation time. In such cases the variables to be checked by the processing loop will be either a counter placed inside the loop or the clock. The case structures on the main driver select the control of the processing loop according to the type of network and user specification of the run duration.

If the run duration is specified by simulated time, and no interval for statistics was defined, a correction has to be done for computation of the average number of jobs in the system. As the simulator timing mechanism is actually controlled by the occurrence of events, the execution of the processing loop terminates at the event which occurs closest to the specified simulated time, see Figure 3.5 . As the computation of the mean number of jobs is time weighted, as explained in Change #5, the last area to be accumulated in this case is calculated from equation 3.11 where  $t_q$  is the last event processed before the simulation time is over.

$$A = (\text{simulated\_time} - t_i) * N_i$$

(eqn 3.11)

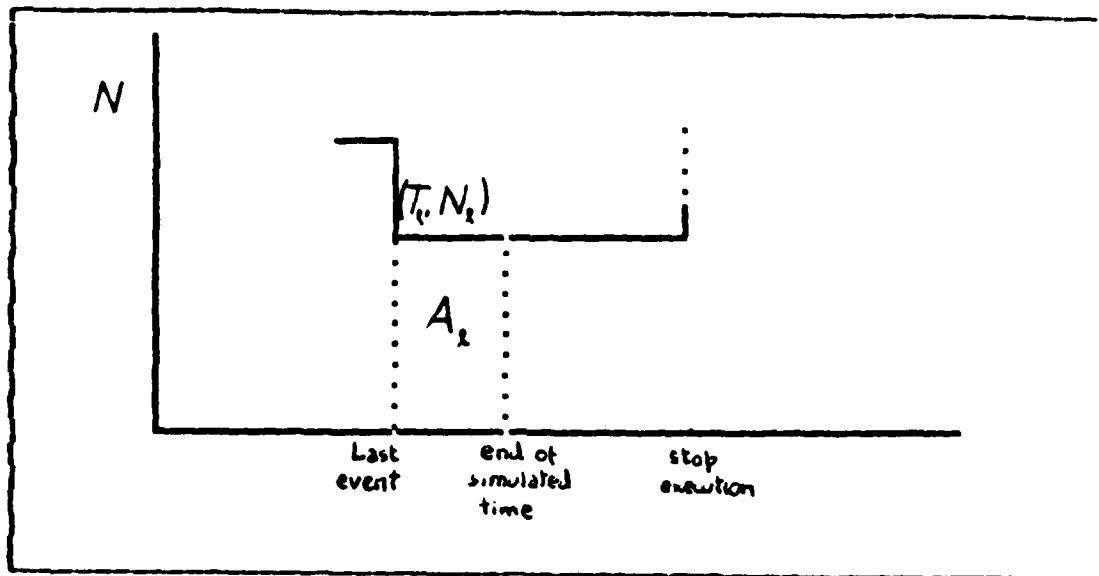


Figure 3.5 Correction of the Accumulated Area

This computation is performed by the procedure STATS\_FCE\_JOBS for all jobs and by the procedure STATS\_FOR\_JOB\_TYPES for each job type.

#### 7. Capability for Rerunning a Simulation

a. CHANGE #9

b. Change Design

The program cycles through the simulation execution code under user control.

c. File affected

CPMT.PAS



d. Module affected

Main Driver

e. Explanation

When a model specification is correct, it can be repeatedly executed. The condition for loop termination is set by the user response to a prompt message.

8. Display of Model Specifications

a. CHANGE #10

b. Change Design

An additional option was included in the MASTER MENU and a new procedure was created for printing single data base records on the screen.

c. Files affected

UFMOD.PAS

CPMT.PAS

MESSAGES.DAT

d. Modules affected

UPDATE

MAIN DRIVER

e. New procedure

DISPLAY\_MODEL

f. Explanation

The new procedure DISPLAY\_MODEL was placed within the UPDATE module and is called from the case construct implemented in the main driver. It first attempts to locate the simulation model in the data base by the

record key computed from the entered simulation model number. If the key is not found an error message is presented, otherwise the record type and number will be prompted for. In this case the procedure attempts to locate the selected record in the data base; if the record key is found the record contents are displayed, otherwise an error message will be produced. The procedure cycles through these steps if the user desires.

#### 9. Printing of Model Specifications

##### a. CHANGE #11

##### b. Change Design

An additional option was included in the MASTER MENU and a new procedure was created for printing all records of a simulation model to the output file.

##### c. Files Affected

UPMOD.PAS

CENT.PAS

MESSAGES.DAT

##### d. Modules Affected

UPDATE

MAIN DRIVER

##### e. New Procedure

PRINT\_DATA\_BASE\_MODEL

##### f. Explanation

The new procedure PRINT\_DATA\_BASE\_model is called from the Main Driver and first attempts to locate the simulation model by the key computed from the entered simulation model number. If the record key is not found an

error message is displayed, otherwise all the records for that simulation model will be printed to the file OUTFILE.

#### 10. Updating of Model Specifications

##### a. .CHANGE #12

##### b. Change Design

Updating of the data base in the original program design is processed by first selecting the option from the MASTER MENU to update the data base, and then entering the simulation model number. If an already existing simulation model number is entered by the user, the program produces an error message, otherwise the UPDATE MENU is displayed.

The new version has a special option in the MASTER MENU to enter a new simulation model number, which will display a list of the simulation numbers already existing in the data base; as the user enters a new simulation model number the UPDATE MENU is automatically displayed and the program becomes ready for record updating.

The update option in the MASTER MENU is to be selected if a user already has simulation models in the data base.

##### c. Files Affected

UPMOD.PAS  
CFMT.PAS  
MESSAGES.DAT

##### d. Modules Affected

UPDATE  
MAIN DRIVER

e. Procedures Changed

ENTER\_SIM\_NUM  
UPDATE\_MENU

f. New Procedure

PRINT\_SIM\_NUM

g. Explanation

The modified procedure ENTER\_SIM\_NUM is called from the MAIN DRIVER rather than from the procedure UPDATE\_MENU, if the options "enter new simulation number" or "updating of model specifications" are selected by the user. If the first option is chosen, the new procedure PRINT\_SIM\_NUM will be called to search for the existing models and display their numbers on the screen. If the data base is empty an appropriate message will be displayed. In both options, but for different reasons, the procedure ENTER\_SIM\_NUM checks for a repeating key before giving access to the UPDATE MENU. Appropriate messages will be displayed for the case of entering a repeated simulation number as a new number, or trying to update a nonexistent simulation model.

11. Deletion and Copy of Simulation Model

a. CHANGE #13

b. Change Design

In the new design, as described in Chapter 2, the scope of the main options is defined at the simulation model level and so copy and deletion of simulation models are options from the MASTER MENU rather than from the UPDATE MENU.

c. Files Affected

UFMOD.PAS  
CENT.PAS  
MESSAGES.DAT

d. Modules Affected

UPDATE  
MAIN DRIVER

e. Procedure Changed

UPDATE\_MENU

f. Explanation

The procedures DEL\_SIM\_MODEL and COPY\_SIM\_MODEL for deletion and copy of model records from the data base were moved outside of the procedure UPDATE\_MENU and are called from the case structure implemented in the main driver.

12. Changing of the Model Specifications

a. CHANGE #14

b. Change Design

Implementation of procedures for modifying the contents of data base records.

c. File Affected

UFMOD.PAS  
MESSAGES.DAT

d. Module Affected

UPDATE

e. Procedure Changed

UPDATE\_MENU

f. New Procedures

CHANGE\_JOB\_TYPE\_REC

CRG\_ROUTING\_REC

CRG\_SERVER\_REC

PRINT\_REC

g. Explanation

The new procedures implemented for changing of data base records are called by the procedure UPDATE\_MENU if the respective option is selected by the user. They control the sequence of events required to compute the correct record key, locate the record, obtain new data and perform changes in the data records. All of the procedures call the new procedure PRINT\_REC to display the contents of the records before and after changes.

13. Handling of Alphabetic Characters

a. CHANGE #15

b. Change Design

The program accepts alphabetic characters as input for options to displayed menus. The characters are converted to integers before selection of code to be executed.

c. Files Affected

UPMOD.PAS

CIET.PAS

d. Modules Affected

UPDATE  
MAIN DRIVER

e. New Function

OPTION

f. Explanation

In the CPMT program design, a set of case structures had been implemented to process the selection of menu options; all the selection variables for these control structures are integers. In the original version the input value which represents the option is an integer and is assigned directly to the selection variable; in the new version the input value is read as a character and converted to integer by the function OPTION, and then is assigned to the selection variable.

14. Printing of Distributions and Queueing Disciplines

a. CHANGE #16

b. Change Design

In order to provide a more comprehensible output, the program distribution type and queueing discipline codes are translated to english before printing on screen and output file.

c. File Affected

UPMOD.PAS

d. Module Affected

UPDATE

e. Procedure Changed

PRINT\_R

f. New Procedure

CCNVERT\_OPT\_STR

g. Explanation

Before printing either on screen or output file the job type and routing records, and for the data items distribution type and queueing discipline, the respective printing procedures call the new procedure CONVERT\_OPT\_STR to convert integer values to preassigned string values.

C. CORRECTIVE CHANGES

1. Error in Deleting a Simulation Model

a. CHANGE #17

b. File Affected

UPMOD.PAS

c. Module Affected

UPDATE

d. Procedure Changed

DELETE\_SIM\_MOD

e. Explanation

The original program terminates if the last simulation model in the data base is deleted. The error was located in the procedure CHECK\_SIM\_SPECS, and it was fixed by adding the end of file (EOF) function, as a condition to be checked in the while loop implemented to delete records.



## 2. Error as Executing a Simulation

a. CHANGE #18

b. File Affected

CHECKSS.PAS

c. Module Affected

CHECK SIM SPECS

d. Procedure Changed

CHECK\_SIM\_SPECS

e. Explanation

The original program terminates if it attempts to run a simulation model with no records stored in the database. The run time error was fixed by calling the procedure that checks errors in the routing record specifications only if no other errors were found in the head or job type records specification.

## 3. Incorrect Handling of Multiservers

a. CHANGE #19

b. File Affected

EXT.PAS

c. Module Affected

EXECUTE AND TABULATE

d. Procedure Changed

FIND\_NEXT\_EVENT\_TIME

e. Explanation

The algorithm to find the next event for a servergroup did not work properly if there is more than one server specified for that servergroup; the results are incorrect statistics for all jobs and individual job type. The error was located in the procedure FIND\_NEXT\_EVENT\_TIME, and it was fixed by adding a test condition to be checked in the loop that searches for the next server to be processed.

D. TYPE AND VOLUME OF CHANGES

The relationship between the type of change performed in the CFMT program and its impact in terms of addition and updating of procedures is illustrated in Table I

**TABLE I**  
**Type and Effect of Changes**

TYPE OF CHANGE	IMPACT ON THE PROGRAM PROCEDURES			
	NEW	MAJOR CHANGE	MINOR CHANGE	TOTAL (%)
ADAPTIVE	13	10	6	29 (64)
PERFECTIVE	4	7	2	13 (29)
CORRECTIVE	-	-	3	3 (7)
TOTAL	17	17	11	

Most of the program modifications were accomplished for development of the simulator modeling capability and program user friendliness. The effect of the work performed to correct errors in the original program is not significant compared to the activity devoted to the satisfaction of new requirements.

#### **E. IMPACT ON THE CPMT USER'S MANUAL**

In order to reflect the enhancement of the CPMT simulator as a result of the changes described in this chapter, rewriting of the CPMT user's manual was required. The next chapter presents the new version of the CPMT user's manual which replaces the original described in Ref.2.

#### **IV. CPMT USER'S MANUAL**

This chapter is intended for CPMT program end users, and includes all the documentation needed to employ the simulator properly. This new version of the user's manual reflects the changes made through the maintenance effort described in chapters 2 and 3, and provides the information users need to prepare a simulation model and run the program.

##### **A. GENERAL DESCRIPTION OF THE CPMT**

CPMT is a network-of-queues simulator designed for simulation of computer systems. The program creates and maintains a database which can store specifications for 99 distinct models. Computer systems are modeled as collections of server groups which represent system components such as CPU and I/O devices.

After modeling the computer system and entering the model parameters in the data base, users can check for correctness of the model specification before running the simulation model. The program includes a built-in debugging aid for simulation design which produces appropriate error messages if the model specification does not meet the established requirements.

Correct simulation models will run for a period of time determined by the user. Upon completion of the simulation run, the program outputs a number of statistics related to the behavior of the simulated system. Users may then study the output and decide whether to rerun the simulation, to change the model parameters or to terminate the program.

A detailed description of the program input , output and possible error conditions is presented in the next sections.

## **B. MODEL DESIGN AND SPECIFICATION**

The specification of a simulation model to be run by the CPMT simulator involves characterizing the computer system configuration and the workload handled by the system.

The system configuration is characterized as a network of hardware resources (CPU, I/O devices or terminals) and software resources (level of programming and scheduling algorithms). The workload which is processed by the system is represented in terms of standard job types, priorities, arrival rates and demands placed on the various resources.

All data parameters to describe the model, except the level of programming, are grouped into three record types for data input and data base storage. The level of programming (number of circulating jobs in a closed network) is not stored in the data base and its specification is entered interactively as the simulation model is run.

Each model is assigned a simulation model number between 1 and 99. The range 1 to 49 is to be used for open network models and the range 50 to 99 for closed network models. The simulation number is used to identify a particular simulation model in the program data base and is common to all the record types developed to describe a given model.

The servergroup record describes the nodes of the computer system being modeled and the job type and routing records describe the work to be processed by the system.

The rest of this section presents a detailed explanation of model design and data input format for simulation of models by the CPMT. An example of the model design process for a simple computer system is provided for better understanding.

## 1. Timing

As the internal clock of the CPMT is an integer, arrival rates and service rates must be represented by integer values. The designer of the simulation model must use (scale up) these time units in a consistent way throughout the model design for correct output statistic results.

## 2. Servergroup Record

Each hardware resource (or node) of the computer system (or network) is described by a servergroup record. Each servergroup is comprised of one or more servers and is serviced by a single queue.<sup>2</sup>

The maximum queue length for the servergroup is assumed to be infinite. The assignment of a job to a server within a servergroup is automatically processed by the program using the following algorithm: servers are assigned to sequential numbers starting by one; a job is assigned to the idle server with the lowest server number.

The servergroup parameters are described below:

SERVERGROUP NUMBER -- the simulator has the capability to model up to 9 distinct servergroups. The user must assign one of the available server group numbers (range 1 to 9).

NUMBER OF SERVERS -- the simulator has the capability to model a maximum of 99 servers within a servergroup. The user specifies the number of servers for each servergroup (range 0 to 99); for each servergroup number not used in the model the user must specify "0" as the number of servers for that servergroup.

---

<sup>2</sup>The order in which the jobs are served is stored in the routing record

### 3. Job Type Record

Modeling of the system workload is done by first partitioning the jobs into classes according to their processing characteristics. Each class is described by a job type record and multiple routing records which are linked to that job type record. The job type data parameters include job type number, job type priority, arrival rate and distribution type and are described below.

JOB TYPE NUMBER -- each job type is assigned a number from 1 to 99 for purposes of identification. The program assigns sequential job type numbers as the job type record data are entered.

JOB TYPE PRIORITY -- for each job type the user specifies the priority which that job will have in the system. The priority range is from 1 to 10 with 1 being the highest. Specification of different job type priorities is insignificant if the jobs are to be served at the servergroups with a nonpriority dependent queueing discipline.

ARRIVAL DISTRIBUTION TYPE AND DISTRIBUTION PARAMETER -- in order to describe the job type arrival rate into the system, the user selects the distribution type and distribution parameter. These parameters are not entered for closed network models (because in a closed network there are no departures or arrivals, and in order to model this, CPMT automatically schedules one arrival at exactly the time of every departure). The distribution type and distribution parameter options are discussed in more detail later in this chapter.

#### 4. Routing Record

Each routing record represents one step in the routing of a job type and has two functions, to describe the service to be processed at each servergroup and to provide the branching probabilities from that servergroup to all the other servergroups in the system.

In order to model the entrance and exit of jobs into the system, two dummy servergroups, 0 and 10, must be described as part of model design. However, no service is performed in these servergroups and so there is no specification of server records for these servergroups. The entrance and exit servergroups must, however, be included in the branching probabilities. The routing record parameters are :

SERVICE DISTRIBUTION TYPE and DISTRIBUTION PARAMETER -- the service demand for the job type is defined by a distribution type and a corresponding parameter. The detailed discussion of these parameters is provided later in this chapter under a separate header.

QUEUEING DISCIPLINE<sup>3</sup> -- the queueing discipline in which the job type is served is identified by an integer value between 1 and 7. The queueing disciplines currently implemented in the CPMT and the corresponding identification code are listed in Figure 4.1.

ROUTING PROBABILITIES -- the routing probability is implemented as a one dimensional array of integers. The entries in this array represent the probability that the particular job type will go from the current server group to each of the other server groups in the model. The routing probability is an integer from 0 to 100. The

---

<sup>3</sup>the queueing discipline is stored in the routing record rather than in the servergroup record for CPMT design convenience.



CODE	QUEUEING DISCIPLINE	ABREV.
1	First Come First Served	(FCFS)
2	Last Come First Served	(LCFS)
3	Nonpreemptive Priority	(NP)
4	Shortest Proc. Time First	(SPTF)
5	Lowest Weighted Proc.Time First	(LWPTF)
6	Processor Sharing	(PS)
7	Served in Random Order	(SIRO)

Figure 4.1 Queueing Disciplines

routing record design must meet established requirements which are discussed further in the "routing design rules".

#### 5. Distribution Specification

To describe the arrival rate of job types or their service rates at the servergroups, the user selects one of three available distribution types, DETERMINISTIC, EXPONENTIAL or UNIFORM, and also provides a parameter (range 1 to 99999) to specify the value(s) of the rate. The distribution types and corresponding distribution parameters are listed in figure 4.2.

#### 6. Routing Design Rules

The routing design for each job type must satisfy the following rules:

- a routing record is required for the entrance servergroup (SG 0). Since no processing is done at this

<u>CODE</u>	<u>DISTRIBUTION</u>	<u>PARAMETER</u>
1	DETERMINISTIC	DETERMINATE VALUE
2	EXPONENTIAL	MEAN VALUE
3	UNIFORM	UPPER BOUND

**Figure 4.2 Distribution Types and Parameters**

server, no values are assigned to the service distribution type or distribution parameter, but routing probabilities from this servergroup to the working servergroups (SG 1 through 9) must be provided.

- jobs must be routed to the exit servergroup (SG 10) from at least one working servergroup; no routing record is required for the exit servergroup because no processing is done at this servergroup and because a job is not routed to any other servergroup.

- the sum of the routing probabilities from a given servergroup to all others must be equal to 100.

- the probability of routing a job from a given servergroup to itself must not be equal to 100 to avoid generating a job which never complete processing.

- if a job type is routed to a servergroup, then a routing record must exist for that job type from that servergroup.

## 7. Data Input Format

In order to facilitate the online input of the model data specification and provide documentation of the model, the user should fill out one servergroup record data form shown in Fig 4.3 per simulation model, and one job type and routing record data form, Fig 4.4 , for each job type in the model.

Simulation number :	
Server Group Number :	Number of Servers :
1	
2	
3	
4	
5	
6	
7	
8	
9	

Figure 4.3 Servergroup Record Data Form

## 8. Example of Model Design

An illustration of the model design process is presented below, using a computer system described by SAUER [Ref. 1 :p.376]. Part of the model specifications will also be used as an example for the user program dialogue described in the next section "HOW TO RUN THE SIMULATOR".

Simulation Number :		Job Type Number :								
*****		*****								
JOB TYPE RECORD										
*****										
Arrival Dist :										
Dist Param :										
Priority :										
*****		*****								
ROUTING RECORD										
*****										
Servergroup :	0	1	2	3	4	5	6	7	8	9
Service Dist:										
Dist Param :										
Queue Disc :										
Routing To :										
SG 1										
SG 2										
SG 3										
SG 4										
SG 5										
SG 6										
SG 7										
SG 8										
SG 9										
SG 10										

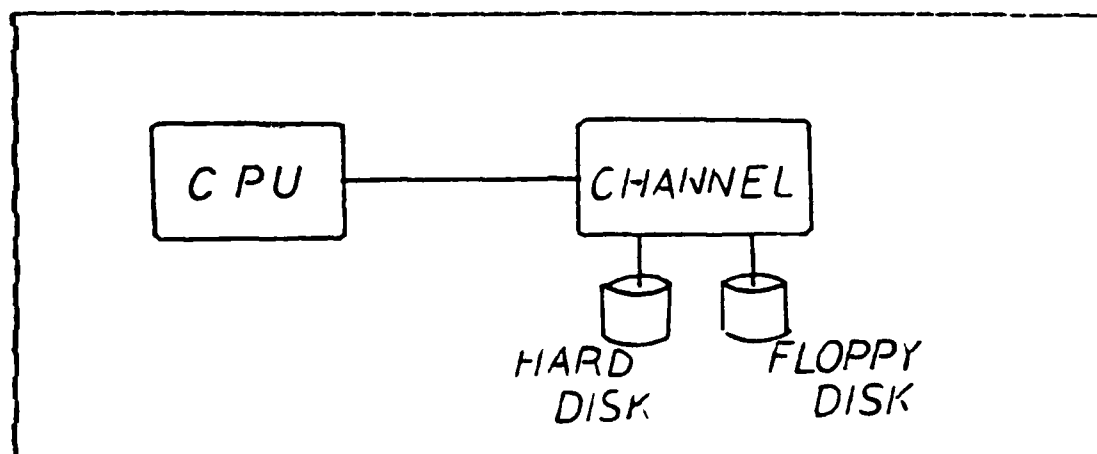
**Figure 4.4 Job Type and Routing Record Data Form**

This model was also simulated for validation of CPMT and the results are discussed in Chapter 5.

**a. The System**

The computer which is to be modeled is a multiprogrammed system having four memory partitions. The system has a single processor and two I/O devices, a floppy

disk and a hard disk, sharing a common channel. The hardware organization is illustrated in Figure 4.5.



**Figure 4.5 Computer System H/W Organization**

The CPU scheduling algorithm is a low overhead Round Robin which can be considered to be equivalent to Processor Sharing (PS) and the I/O requests are served in a First Come First Served basis. The system is to be simulated with all memory partitions in use. The degree of multiprogramming, average service times and branching probabilities derived from the system accounting data recorded during a period of heavy workload are illustrated in Figure 4.6.

#### **1. The Model**

Assuming that there is a sufficient backlog of jobs and there is a sufficient memory contention that the degree of multiprogramming is essentially constant, the system can be modeled by a closed queueing central server network model. The central processor is represented by servergroup 1 preceded by a queue, and the disks are represented as servergroups 2 and 3, each one with a

DEGREE OF MULTIPROGRAMMING .....	4
AVERAGE CPU SERVICE TIME .....	0.05 sec
AVERAGE FLOPPY DISK SERVICE TIME .....	0.220 sec
AVERAGE HARD DISK SERVICE TIME .....	0.019 sec
PROBABILITY OF JOB COMPLETION	
AFTER DISK SERVICE .....	0.125
PROBABILITY OF FLOPPY ACCESS .....	0.1

Figure 4.6 Data Parameters

respective queue. Servergroups 2 and 3 are organized in parallel with respect to the central processor. Two additional dummy servers, entrance (SG 0) and exit (SG 10) are included as required by the CPMT. The model is illustrated in Fig 4.7.

Service times are assumed to be exponentially distributed.

#### c. Input Model Parameters

As the system is represented by a closed network, a simulation number in the range 50 to 99 must be assigned to the simulation model. For this example the simulation number 60 was arbitrary chosen.

SERVERGROUP RECORD -- the model has three servergroups SG1 (CPU), SG2 (FICPPY DISK) and SG3 (HARD DISK) each one consisting of a single server. The servergroup record data form for this model is displayed in figure 4.8.

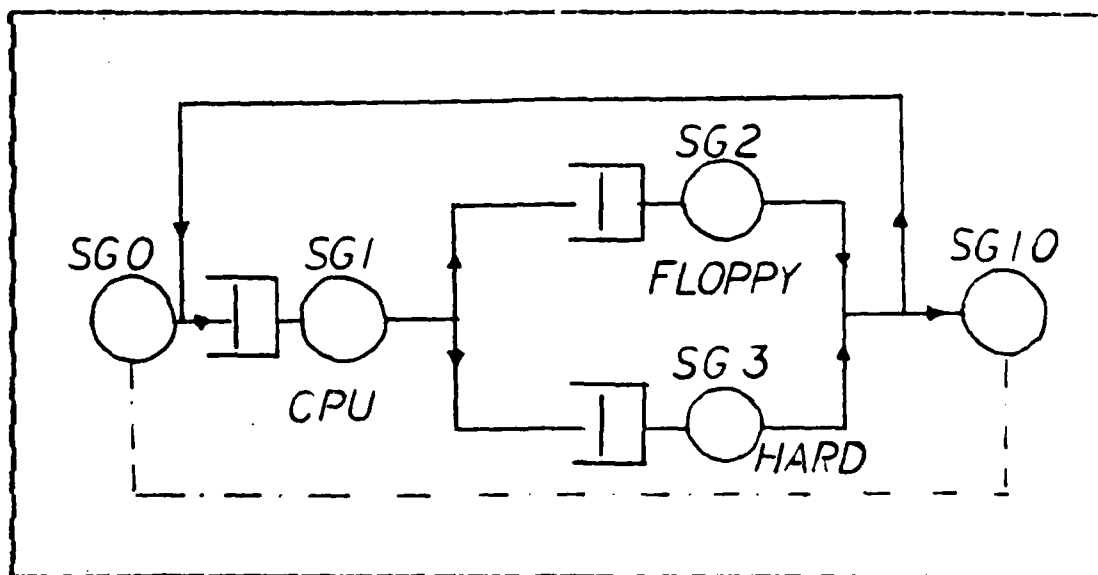


Figure 4.7 Computer System Model

Simulation number : 60

Server Group Number :	Number of Servers :
1	1
2	1
3	1
4	0
5	0
6	0
7	0
8	0
9	0

Figure 4.8 Model Servergroup Data Form

JOB TYPE RECORD — the computer system has only one job type which will be designated as job type 1, and since there is only one, the priority is insignificant. As the model is a closed network, arrival distribution type and distribution parameter are not considered.

ROUTING RECORDS -- three routing records are required to describe the service processed at the servergroups and routing of jobs through the system. As stated in the routing design rules an additional servergroup (entrance servergroup) record is required for job routing purpose.

The smallest amount of time represented in the model, as listed in Fig 4.6 is the hard disk service, which is 0.019. Therefore, all the time values are multiplied by 1000 so that they are all integers (time unit = millisecond). The mean service times are then :

SG 1 .....	50
SG 2 .....	220
SG 3 .....	19

The routing probabilities to be assigned are derived from data shown in Fig 4.6 , applying the routing design rules for CPMT. As the processing of jobs is initiated by CPU service, the routing probability from the entrance servergroup SG0 to SG1 is 100. The routing probability from SG1 to SG2 is 10 and from SG1 to SG3 is 90 (100 - 10). As the probability of job completion after disk service is 0.125, the rounded value in the range 0 to 99 is 13. Thus the routing probabilities from SG2 and SG3 to the exit servergroup SG10 are 13. The remaining routing probability (for new CPU service) is 87, thus the routing probabilities from SG2 and SG3 to SG1 are 87. The job type and routing records data form for the model are illustrated in Fig 4.9.



```

Simulation Number : 60           Job Type Number : 1
***** JOB TYPE RECORD *****
Arrival Dist : -
Dist Param   : -
Priority      : 1
***** ROUTING RECORD *****
Servergroup : 0  1  2  3  4  5  6  7  8  9
Service Dist: -  2  2  2
Dist Param   : - 50 220 19
Queue Disc   : -  6  1  1
Routing To   :
    SG 1      100  0  87  87
    SG 2       0 10  0  0
    SG 3       0 90  0  0
    SG 4       0  0  0  0
    SG 5       0  0  0  0
    SG 6       0  0  0  0
    SG 7       0  0  0  0
    SG 8       0  0  0  0
    SG 9       0  0  0  0
    SG 10      0  0 13 13

```

Figure 4.9 Model Job Type and Routing Form

### C. HOW TO RUN THE SIMULATOR

CPMT runs on the VAX/VMS Computer Science Department Computer at NPS. To execute the program after logging onto the computer, enter the command RUN CPMT in response to the \$ prompt.

The program initially displays to the user the MASTER MENU of program options presented in Fig. 4.10 The user enters the integer value corresponding to the option desired. Whenever an invalid option is entered the menu is redisplayed. A description of each option follow under separate headings.

- 1 - ENTER NEW SIMULATION NUMBER
- 2 - UPDATE SIMULATION SPECIFICATIONS
- 3 - CHECK SIMULATION SPECIFICATIONS
- 4 - RUN SIMULATION MODEL
- 5 - PRINT ALL DATA BASE
- 6 - PRINT DATA BASE FOR A SINGLE MODEL
- 7 - DELETE SIMULATION MODEL
- 8 - COPY SIMULATION MODEL
- 9 - DISPLAY SIMULATION MODEL SPECIFICATIONS
- 0 - EXIT CPMT ENVIRONMENT

Figure 4.10 Master Menu Options

At several points in the program, the user directs program control by responding to questions which have "yes" or "no" answers. The convention for the CPMT program is that the user enters either the uppercase or lowercase "y" for a "yes" response and any other character for a negative response.

Each simulation model is identified by a unique integer value called the simulation number. The user may assign a simulation model an integer number in the range 1 to 49 for OPEN NETWORK MODELS and 50 to 99 for CLOSED NETWORK MODELS.

## 1. Enter a New Simulation Number

This function is to be selected if the user wishes to add a new simulation model to the simulator data base.

Upon entry of the integer value "1" corresponding to this option the program displays either the simulation numbers already existing in the data base, or the message :

NO SIMULATION NUMBERS IN DATA BASE

and prompts for entering of the simulation number. At this point the user enters the desired simulation number for the new model. If a simulation number out of the 1 to 99 range is entered, the message

ERROR IN INPUT

is displayed and the simulation number is prompted again. Otherwise, if the user enters a simulation number already existing in the data base, the message

SIMULATION MODEL NUMBER ALREADY EXISTS IN DATA BASE

is displayed and the program will return to the Master Menu on the entry of any character. Otherwise the update options are presented to the user in a menu format, UPDATE MENU, similar to that of the MASTER MENU. The update menu options are listed in Fig 4.11 and are explained further in this section under a separate header.

## 2. Update Model Specifications

This function is to be selected if the user wishes to update the specifications of a simulation model already existing in the simulator data base. This function is also automatically selected after a new simulation number has been selected.

- 1 - ADD JOB TYPE RECORD
- 2 - ADD ROUTING RECORD
- 3 - ADD SERVERGROUP RECORD
- 4 - DELETE JCB TYPE RECORD
- 5 - DELETE ROUTING RECORD
- 6 - DELETE SERVERGROUP RECORD
- 7 - CHANGE JCB TYPE RECORD
- 8 - CHANGE ROUTING RECORD
- 9 - CHANGE SERVERGROUP RECORD
- 0 - RETURN TO MASTER MENU

Figure 4.11 Update Menu Options

Upon entry of the integer value 2 corresponding to this option, the program prompts the user to input the simulation number. If the user enters a simulation number that does not exist in the data base the following message is displayed :

SIMULATION MODEL DOES NOT EXIST IN DATA BASE

In this case program control returns to the Master Menu after entry of any character, otherwise the update options are presented by the UPDATE MENU.

### 3. Check Simulation Specifications

This option is used as a debugging aid to check if a given simulation model meets the specification requirements for a successful simulation run.

The program will prompt the user to input the number of the simulation model to be checked. As the user enters

the simulation model number, the existence of data records and observation of the routing rules are tested for the specified model. If the simulation model does not meet the requirements the following message is displayed :

**SIMULATION SPECIFICATIONS DID NOT CHECK  
ERROR MESSAGES IN FILE OUTFILE.DAT**

The error messages will help the user to eliminate the model specification deficiencies. The list of possible error messages is presented in Figure 4.12.

1. Simulation Number Does Not Exist.
2. No Server Group Record Exists.
3. No Job Type Exists.
4. Job Numbers Are Not Sequential.
5. Server Group\_\_, Job Type\_\_ Routing Loop.
6. No Routing Records Exist for Job Type\_\_.
7. No Server Group 0 Routing Record For Job Type\_\_.
8. Job Type\_\_ not Routed to Exit Server Group.
9. Job Type\_\_ Routed To But Not From Server Group\_\_.

**Figure 4.12 Simulation Specification Error Messages**

If the simulation model is correctly specified the following message is displayed :

**SIMULATION SPECIFICATIONS CHECK**

In both cases the program control returns to the Master Menu by entry of any character.

#### 4. Run a Simulation Model

This option is selected if the user wishes to execute the simulation of a model.

The program will first prompt the user to input the number of the model to be run and then displays a menu with the options for specification of run duration, number of jobs, clock time or number of events ( or the last two options for closed network models). Upon entering the option the program displays the prompt for entering of the corresponding parameter. As the user enters the number of jobs, simulation time or number of events to be processed, depending on the specification type selected, the program prompts for a seed value. The seed will be used as initial input into the system random number generator. The random numbers in turn are used as input into program functions which require random variables.

At this point the program asks if the user wishes the program to specify the interval of time for statistics (the program can produce statistics about the behavior of the simulated system for a period of time during simulation or for all simulation time). If the user response is affirmative, the start time and stop time for gathering statistics will be requested.

If the simulation model to be run is a closed network, the program will ask the user to input an additional model specification, the degree of programming. The degree (or level) of programming represents the number of jobs to be processed in the closed network and is prompted for each job type in the model. A complete example of the user program dialogue for execution of a closed network simulation model is illustrated in Fig. 4.13.

Before executing the simulation model the program will check the simulation specifications. If the model is

not correctly specified the following message will be displayed :

SIMULATION MODEL NOT EXECUTED  
SIMULATION MODEL SPECIFICATIONS DO NOT CHECK  
ERROR MESSAGES IN FILE OUTFILE.DAT

```
ENTER SIMULATION NUMBER OF MODEL TO EXECUTE
65
ENTER SPECIFICATION TYPE FOR RUN DURATION
                                1- CLOCK TIME
                                2- NUMBER OF EVENTS
1
ENTER SIMULATION RUN DURATION
150000
ENTER THE SEED YOU WANT TO USE
45367
DO YOU WISH TO SPECIFY THE INTERVAL TIME FOR GATHERING
STATISTICS ? y/-
Y
ENTER START TIME FOR STATISTICS
300
ENTER STOP TIME FOR STATISTICS
120000
ENTER DEGREE OF PROGRAMMING OF JOB TYPE 1
4
```

**Figure 4.13    Example of Execute Simulation Model Dialogue**

The possible error messages are those already referenced and illustrated in Figure 4.12. Otherwise if the simulation run duration is selected by clock time and no jobs complete

within the simulation period, the following error message will be displayed :

```
ERROR - SIMULATION MODEL EXECUTED BUT  
NO JOBS COMPLETED DURING SIMULATION TIME
```

In both cases the program returns to Master Menu by entry of any character.

If the execution is successful the statistical output will be written in the file OUTFILE.DAT and the following message will be displayed :

```
SIMULATION MODEL EXECUTED  
OUTPUT STATISTICS IN FILE OUTFILE.DAT  
DO YOU WISH TO RUN THE SIMULATION AGAIN ? y/-
```

If an affirmative response is entered the program dialogue will be repeated for rerunning the same simulation model, otherwise the user is given the option of exiting the function or attempting to run other simulation model.

The output statistics include minimum, maximum, mean and standard deviation of time in system and time in queue, throughput and mean number of jobs in the system for all jobs and for each job type, and maximum, minimum and mean queue size and utilization by server group and server within a server group. An example of the output report format is illustrated in Figure 4.14.



```

SIMULATION NUMBER : 60
SIMULATION TYPE : CLOSED NETWORK
SEED 45367
START STATISTICS TIME: 300
STOP STATISTICS TIME: 120000
NUMBER OF EVENTS: 6213
SIMULATION TIME: 150000
JOB TYPE MAX QTIME MIN QTIME MEAN QTIME STDD QTIME MAX STIME MIN STIME MEAN STIME STDD STIME MEAN THROUGH PUT
ALL
1 ( data )
SERVER GROUP MAX QLEN MIN QLEN AVG QLEN SG UTIL SERVER NUMB SERVER UTIL
1
2 ( data )
3

```

Figure 4.14 Example of Statistical Output Report

## 5. Print Data Base

This option is used for monitoring of the CPMT data base workload.

Upon selection of this option, a printout of the entire indexed sequential data base is written to the file OUTFILE.DAT. The program control returns to Master Menu by entering of an arbitrary character. If the user wishes a hard copy a print out must be requested from outside the CPMT environment.

## 6. Print Data Base for a Single Model

This function is to be selected if the user wishes a printout of a given simulation model specification.

Upon selection of this option, the program asks the user to input the simulation number. Upon entry of the simulation model number, the program attempts to find the simulation model in the data base. If the simulation model exists, the program writes all the records for that model to the file OUTFILE.DAT, and displays the message:

MODEL SPECIFICATION IN FILE OUTFILE.DAT

The user then requests a printout of the file from outside the CPMT environment. If the simulation model number is not found the message

SIMULATION MODEL DOES NOT EXIST IN DATA BASE

is displayed. In either case the user is given the option of exiting this function, or attempting to print another simulation model.

## 7. Delete a Simulation Model

This option is used to delete a complete simulation model from the data base.

Upon selection of this option the program prompts for entry of the simulation model number. After the user enters the simulation number the program attempts to find the model in the data base. If the number of the model does not exist an error message is displayed, otherwise the program gives the user the option of deleting the model. If the user responds affirmatively the program deletes all of the records for the chosen simulation model from the data base and displays the message:

SIMULATION MODEL \_\_ DELETED.

At the end of the dialogue the user is given the option of exiting function or attempting to delete other simulation model.

#### 8. Copy a Simulation Model

The copy option is convenient if the user wishes to compare the simulation results of two models with a few changes in parameter specifications. In this case the user can copy one model to a new number, make the changes in the copy, and maintain both model design specifications in the data base.

Upon selection of this option the program displays the prompt for entering the simulation model number which is to be copied and the model number of the copy. If the number of the model to be copied does not exist the following message is displayed :

SIMULATION MODEL NUMBER \_\_ DOES NOT EXIST IN DATA BASE

Otherwise, if the new simulation model number is already in the data base the following message is displayed :

SIMULATION MODEL NUMBER \_\_ ALREADY EXISTS IN DATA BASE

If the copy is successful the following message is displayed

SIMULATION MODEL COPIED

At the end of the dialogue the user is given the option of exiting the function or attempting to copy another simulation model.

#### 9. Display Simulation Model Specifications

This option is used for on-line review of simulation models.

Upon selection of this option the program prompts for entry of the simulation model number and then attempts to find the model in the data base. If the simulation model does not exist an appropriate message will be displayed, otherwise the program asks the user to input the type of record to be reviewed (job type ,routing or servergroup). If the user selects either the job type or the routing record, the program asks the user to identify the job type number. The record data will be directly displayed for the job type record and the identification of the routing record number to be displayed will be prompted for the routing record option. If a servergroup record is selected for user review the program asks the user to identify the servergroup number and then displays the record data.

For all the options the program displays error messages if the user attempts to display data from nonexistent records.

After record data review the user is given the option of displaying another record for the same model. If a negative response is entered, the user is given the option of exiting the function, returning to the Master Menu options or displaying another model.

The user program dialogue for displaying a routing record is shown in Figure 4.15.

```

*****
      DISPLAY SIMULATION MODEL FUNCTION
*****

ENTER SIMULATION NUMBER YOU WISH TO DISPLAY DATA
60

ENTER RECORD TYPE YOU WISH TO DISPLAY

                                1- JOB TYPE
                                2- ROUTING
                                3- SERVERGROUP

2

ENTER JOB TYPE NUMEER
1

ENTER ROUTING RECOED NUMBER
3

      (clear screen)
      (display of function header)

      SERVICE DISTRIBUTION IS      : EXPONENTIAL
      DISTRIBUTION PARAMETER IS    : 19
      QUEUEING DISCIPLINE IS      : FCFS

      ROUTING PROB TO 1 IS : 87
      ROUTING PROB TO 2 IS : 0
      ROUTING PROB TO 3 IS : 0
      ROUTING PROB TO 4 IS : 0
      ROUTING PROB TO 5 IS : 0
      ROUTING PROB TO 6 IS : 0
      ROUTING PROB TO 7 IS : 0
      ROUTING PROB TO 8 IS : 0
      ROUTING PROB TO 9 IS : 0
      ROUTING PROB TO 10 IS : 13

DO YOU WISH TO DISLAY MORE RECORDS ? Y/-
N

DO YCU WISH TO EXIT FUNCTION ? Y/-
Y

```

Figure 4.15 Example of Display Simulation Model Dialogue

#### 10. Exit CPMT Environment

Upon selection of this option the program execution terminates and control returns to the system

## 11. Updating Options

The functions for data base record updating that are presented by the UPDATE MENU are related to a simulation model number already entered by the user after either selecting the option "Enter New Simulation Number" or "Update Model Specifications" from the MASTER MENU. The Record Updating functions are explained below.

### a. Add Job Type Record

Upon selection of this option the program automatically accesses the simulation model in the data base to determinate the next available job type number for the given simulation model and assigns that number to the job type to be added. The program then requests the user to input the arrival distribution and distribution parameters,\* and priority of the job type. Input data values are echced as they are entered, to allow the user to detect incorrect data. The program also prompts for re-input of invalid data.

As the job type record data is entered the program displays the entries for review and asks if the user wishes to add the job record. If the user chooses to add a job type record which exists in the data base, the program responds with the message

RECORD ALREADY EXISTS, NOT ADDED

otherwise the job type record is added to the data base and the message

RECORD SUCCESSFULLY ADDED

---

\*These two parameters are not requested for closed network models

is displayed. If the user chooses not to add the record the program displays

RECORD NOT ADDED

If the job type record is successful added the option of going directly to the Add Routing Record function (see next option) is provided. The program control will return to the add job type function upon exit from the add routing record function.

The user may add multiple job type records to a model. At the end of every iteration of the option dialogue the user is given the choice of exiting the function or adding another job type to the model. The dialogue for addition of a job type record is illustrated in Fig 4.16.

#### b. Add Routing Record

This option can be entered either directly from the add job type function as explained above, or from the update menu. When the user selects this option from the add job type function, the routing records are automatically added to the job type record just added. Otherwise the program will ask the user to identify the job type number for which routing records are to be added. If the job type number is not found in the data base the following error message is displayed:

ERROR THE JOB TYPE RECORD DOES NOT EXIST

If the record job type is found the program requests the user to input the routing parameters of servergroup number, service distribution, distribution parameter, queueing discipline and routing probabilities.

As in the "add job type function" the input data and the entire data record are displayed for user review. The user is given the option of adding the routing record.

```

*****
      ADD JOB TYPE RECORD FUNCTION
*****
SIMULATION MODEL NUMBER: 60 *
JOB TYPE NUMBER:1
ENTER PRIORITY FOR THIS JOB TYPE:
VALID PRIORITY CODES ARE 1 THRU 10
1
      (clear screen)
      (display of function header)
      (display of the job type record)
DO YOU WISH TO ADD RECORD TO THE DATA BASE ? Y/-
Y
      RECORD SUCCESSFUL ADDED
DO YOU WISH TO ADD ROUTING RECORDS FOR THIS JOB TYPE?
N
DO YOU WISH TO EXIT FUNCTION ? Y/-
Y
* As the model is a closed queueing network
  arrival distribution type and distribution parameter
  are not prompted

```

Figure 4.16 Add Job Type Record Dialogue

If the user chooses to add an existing routing record, the record is not added and an error message is displayed.

At the end of the function dialogue the user has the option of exiting the function or adding another routing record to the model.

An example of the user program dialogue for adding a routing record with selection from the UPDATE MENU is displayed in Figure 4.17.



```

*****
      ADD ROUTING RECORD FUNCTION
*****

SIMULATION MODEL NUMBER: 60

ENTER JOB TYPE NUMBER OF ROUTING RECORDS TO BE ADDED
1

ENTER ROUTING RECORD SERVER GROUP NUMBER:
2

ENTER SERVICE RATE DISTRIBUTION: 1-DETERMINATE
                                2-EXPONENTIAL
                                3-UNIFORM
2

ENTER EXPONENTIAL DISTRIBUTION MEAN:
220

ENTER QUEUEING DISCIPLINE:
1-FIRST COME FIRST SERVED      (FCFS)
2-LAST COME FIRST SERVED      (LCFS)
3-NONPREEMPTIVE PRIORITY      (NP)
4-SHORT. PROC. TIME FIRST     (SPTF)
5-LOW. WEI. PROC. TIME FIRST  (LWPTF)
6-PROCESSOR SHARING           (PS)
7-SERVICE IN RAND. ORDER     (SIRO)

1

ENTER THE ROUTING PROBABILITY FROM SERVER GROUP 2 TO
SERVER GROUP 1: 87
SERVER GROUP 2: 0
SERVER GROUP 3: 0
SERVER GROUP 4: 0
SERVER GROUP 5: 0
SERVER GROUP 6: 0
SERVER GROUP 7: 0
SERVER GROUP 8: 0
SERVER GROUP 9: 0
SERVER GROUP 10: 13

      (clear screen)
      (display of function header)
      (display of the routing record)

DO YOU WISH TO ADD RECORD TO THE DATA BASE ? Y/-
Y

      RECORD SUCCESSFUL ADDED

DO YOU WISH TO EXIT FUNCTION ? Y/-
Y

```

Figure 4.17 Add Routing Record Dialogue

### c. Add Servergroup Record

This option is used to add a servergroup record to a simulation model.

Upon selection of this option the program prompts the user for entering the number of servers for each server group in the system and then displays the record for user review. At this point the user is given the option of adding the record to the data base. If the user chooses not to add the record, the message:

RECORD NOT ADDED

is displayed, otherwise the program will check for the existence of a servergroup record for that model in the data base before adding the new record. Depending on whether there is an existing servergroup record for that model, the record is added or not and one of the following messages is displayed :

RECORD SUCCESSFULLY ADDED

or

RECORD ALREADY EXIST; NOT ADDED

This function is not repeated because there is only one allowed servergroup record per simulation model, and the user is returned to the Update Menu on entry of a character.

An example of the user program dialogue for adding a servergroup record is illustrated in Fig. 4.18.

### d. Delete Job Type Record

This option is used to delete a job type record from the data base.

Upon selection of this option the program requests that the user input the job type number of the job

```

*****
ADD SERVER GROUP RECORD FUNCTION
*****

SIMULATION MODEL NUMBER: 60

ENTER NUMBER OF SERVERS FOR

SERVER GROUP 1: 1
SERVER GROUP 2: 1
SERVER GROUP 3: 1
SERVER GROUP 4: 0
SERVER GROUP 5: 0
SERVER GROUP 6: 0
SERVER GROUP 7: 0
SERVER GROUP 8: 0
SERVER GROUP 9: 0

      (clear screen)
      (display of function header)
      (display of the servergroup record)

DO YOU WISH TO ADD THIS RECORD ? Y/-
Y

      RECORD SUCCESSFUL ADDED

ENTER ANY CHAR TO RETURN TO UPDATE MENU

```

Figure 4.18 Add Servergroup Record Dialogue

type record to be deleted. If the job type does not exist in the data base the following message is displayed:

NC RECORD FOUND

otherwise the record is displayed and the user is given the option of deleting it from the data base. Depending on the user response the record is deleted or not and one of the following messages is displayed:

RECORD SUCCESSFULLY DELETED

OR

RECORD NOT DELETED

At the end of the function dialogue the user has the option of exiting function or attempting deletion of another job type record for the same simulation model.

WARNING :

WHEN A JOB TYPE RECORD IS DELETED ALL THE ROUTING RECORDS WHICH ARE SUBORDINATED TO THAT JOB TYPE ARE ALSO DELETED

e. Delete Routing Record

This option is used to delete routing records of a simulation model.

Upon selection of this option the program requests that the user input the job type number and servergroup number to which the routing record is attached. If either the job type record or routing record are not found in the data base an error message is displayed, otherwise the user is given the option of deleting the specified record. Depending on the user option the record is deleted or not and one of the following messages is displayed:

RECORD SUCCESSFULLY DELETED

OR

RECORD NOT DELETED

At the end of the option dialogue the user is given the option of exiting the function or deleting another routing record.

#### f. Delete Server Group Record

This option is used to delete the server group record from the data base.

Upon selection of this option the program attempts to locate the server group record in the data base. If the record does not exist an error message is displayed, otherwise the record is deleted and the message

RECORD SUCCESSFULLY DELETED

is displayed. As there is only one server group record per simulation model, at the end of the option dialogue user returns to Update Menu options by entering an arbitrary character.

#### g. Change Job Type Record

This option is used to change model parameters stored into a job type record.

The program first requests that the user input the job type number of the job type record to be changed. If the record does not exist the following message is displayed:

CHANGE ERROR JOB TYPE NUMBER NOT FOUND

otherwise the record is displayed and the user is given the option of selecting the model parameter to be modified, arrival distribution type, distribution parameter, or job priority. As the user option is entered, the program prompts for input data according to the model parameter selected. When the input is complete the user has the option to select another model parameter to be changed. If the user chooses to modify another parameter then the menu of options for changing of model parameters will be displayed for another function iteration. Otherwise the entire updated job type

record will be displayed for user review. At this point the user is given the option of exiting the function or changing another job type record. An example of the user program dialogue for changing a job type record is illustrated in Fig 4.19.

#### h. Change Routing Record

This option is used to change the model parameters stored in a routing record.

The program first requests that the user input the job type number of the job type record to which the routing record to be changed is subordinated and then asks the user to identify the servergroup number of the routing record. If either the job type or routing record are not found in the data base appropriate error messages are displayed, otherwise the user is given the option of selecting the model parameter to be changed, service distribution, service distribution type, queueing discipline or routing probabilities. Upon selection of the model parameter to be changed the program prompts for entering data according to the option selected. When the input of new data is complete the user has the option to change another model parameter. If the user chooses to change another parameter, a new function iteration will be processed for the same routing record, otherwise the routing record is displayed and user is given the option of exiting the function or changing another routing record.

An example of the user program dialogue for changing a routing record is illustrated in Fig 4.20.

#### i. Changing Server Group Record

This option is used to change the server group record for a given simulation model.

```

*****
CHANGE JCB TYPE RECORD FUNCTION
*****

SIMULATION MODEL NUMBER:  5
ENTER NUMBER OF JOE TYPE TO CHANGE:
2
SIMULATION MODEL NUMBER:  5
JOE TYPE NUMBER:  2
    ARRIVAL DISTRIBUTION IS:  EXPONENTIAL
    DISTRIBUTION PARAMETER IS:  50
    JCB PRIORITY IS:          1
ENTER PARAMETER YOU WISH TO CHANGE
                                1- ARRIVAL DISTRIBUTION
                                2- DISTRIBUTION PARAMETER
                                3- JOB PRIORITY
2
ENTER EXPONENTIAL DISTRIBUTION MEAN
100
DO YOU WISH TO CHANGE OTHER  PARAMETER ? Y/-
N

                (clear screen)
                (display of function header)
SIMULATION MODEL NUMBER IS: 5
JOE TYPE NUMBER IS: 2
    ARRIVAL DISTRIBUTION IS:  EXPONENTIAL
    DISTRIBUTION PARAMETER IS:  100
    JOB PRIORITY IS:          1
DO YOU WISH TO EXIT FUNCTION ? Y/-
Y

```

**Figure 4.19 Changing Job Type Record Dialogue**

Upon selection of this option the program attempts to find the servergroup record in the data base. If

```

*****
CHANGE RCUTING RECORD FUNCTION
*****

SIMULATION MODEL NUMBER: 60
ENTER NUMBER OF JOE TYPE:
1
ENTER NUMBER OF RCUTING RECORD TO CHANGE
1

SERVICE DISTRIBUTION IS : EXPONENTIAL
DISTRIBUTION PARAMETER IS : 19
QUEUEING DISCIPLINE IS : FCFS

ROUTING PROB TO 1 IS : 97
ROUTING PROB TO 2 IS : 0
ROUTING PROB TO 3 IS : 0
ROUTING PROB TO 4 IS : 0
ROUTING PROB TO 5 IS : 0
ROUTING PROB TO 6 IS : 0
ROUTING PROB TO 7 IS : 0
ROUTING PROB TO 8 IS : 0
ROUTING PROB TO 9 IS : 0
ROUTING PROB TO 10 IS : 13

ENTER PARAMETER YOU WISH TO CHANGE
1- SERVICE DISTRIBUTION
2- DISTRIBUTION TYPE
3- QUEUEING DISCIPLINE
4- ROUTING PROBABILITIES
3

ENTER QUEUEING DISCIPLINE
1-FIRST COME FIRST SERVED (FCFS)
2-LAST COME FIRST SERVED (LCFS)
3-NONPREEMPTIVE PRIORITY (NP)
4-SHORT. PROC. TIME FIRST (SPTF)
5-LOW. WEI.PROC.TIME FIRST (LWPTF)
6-PROCESSOR SHARING (PS)
7-SERVICE IN RAND. ORDER (SIRO)
6

DO YOU WISH TO CHANGE OTHER PARAMETER ? Y/-
N

      (clear screen)
      (display of function header)
      (display of the routing record)

DO YOU WISH TO EXIT FUNCTION ? Y/-
N

```

Figure 4.20 Change Routing Record Dialogue



the record is not found an error message will be displayed, otherwise the number of servers for each servergroup will be prompted. As these data are entered the updated servergroup record is displayed for user review. The user is returned to the UPDATE MENU by entry a character.

## 12. SUMmary of the Manual Contents

This CPMT user's manual was primarily designed for the CS 4400 students at the Naval Postgraduate School. The first section introduces the simulation program to the new CPMT users. As the users should be comfortable with the model design before CPMT utilization, the next section of the manual describes and illustrates with an example the design of simulation models to be run by CPMT. The last section explains how to store and update model specifications in the simulator data base, and run a simulation. Some examples of the user program dialogue are displayed for easier familiarization with the simulator.

## V. TESTING AND VALIDATION

The objective in this chapter is to validate the CPMT ability to simulate computer systems. To accomplish this validation a set of testing models was run and the output results analyzed. The characteristics of the testing models include CPMT capabilities not analyzed by Lt Pagel [Ref. 2] and additional enhancements implemented through this thesis effort.

The criteria and procedures used for validation are described in the first section, followed by a description of each experiment and an interpretation of the collected data. A summary of conclusions is presented at the end of the chapter.

### 1. Criteria

The overall criterion used for CPMT validation was that the conclusions that are drawn from running a simulation model on the simulator should be the same conclusions that would have been drawn from evaluating the model in analytical form.

A random sample of 10(n) measurements of each output parameter is collected from independent simulation runs. Based on this sample, a two tailed hypothesis test is performed on the mean value to decide whether the simulation results come from the same population as the analytical results.

Levels of significance of .05 and .01 were established as acceptable accuracy bounds. As the sample is small, it was assumed to come from a student-t distribution with 9 (n-1) degrees of freedom.

The null hypothesis is the following :

AD-A159 003

DEVELOPMENT AND ENHANCEMENT OF COMPUTER PERFORMANCE  
MODELING TOOL (CPMT) (U) NAVAL POSTGRADUATE SCHOOL  
MONTEREY CA J G DUARTE JUN 85

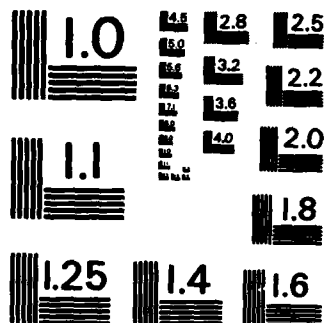
2/2

UNCLASSIFIED

F/G 9/2

NL

									END			
									FROM			
									DATE			



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS - 1963 - A

$$H : \mu = \mu_0$$

where  $\bar{x}$  is the sample mean and  $\mu_0$  is the analytical parameter. The alternative hypothesis is :

$$H : \mu \neq \mu_0$$

The critical values obtained from the t-statistical table for the levels of significance and degree of freedom described above are listed in Table II.

**TABLE II**  
**Critical Values from T-statistical Table**

LEVEL OF SIGNIFICANCE	CRITICAL VALUE
0.05	1.833
0.01	2.821

The sample mean  $\bar{x}$  is transformed to student-t distribution by equation 5.1 , where  $\mu_0$  is the analytical parameter, S the sample standard deviation and n the sample size.

$$t = (\bar{x} - \mu_0) / (S / \sqrt{n}) \quad \text{(eqn 5.1)}$$

For a given sample the null hypothesis is rejected if the statistic computed from equation 5.1 is greater than the critical value for the level of significance being considered. Otherwise the null hypothesis is accepted.

## 2. Test Case #1

### a. Objective

The objective is to test the simulator behavior for a deterministic service rate and independent service demand queueing discipline (FCFS, LCFS or SIRO). The output parameter to be measured is the system throughput.

### b. Simulation Model

The simulation model consists of a single server queueing model in which the arrival rate is exponentially distributed and the service rate is constant ( $M/D/1$ ). The interarrival mean and service mean are 100 and 50 respectively. The model is to be run independently for FCFS, LCFS and SIRO queueing disciplines.

### c. Simulation Results

The simulation run duration was specified by the number of jobs to be processed and the output results are listed in Table III.

### d. Analytic Results

The system throughput for a single server model is equal to the arrival rate. As the interarrival mean for the model is 100 the arrival rate and throughput rate are equal to 0.01.

### e. Statistical Analysis

The mean and standard deviation of the system throughput obtained from the samples are listed in Table IV. The values of the statistic computed from equation 5.1 using the mean and standard deviation listed in Table IV are 1.941, 2.395 and 2.678 for FCFS, LCFS and SIRO queueing disciplines. As these values are less than the critical value

**TABLE III**  
**Simulation Results of Test Case #1**

RUN #	NUMBER JOBS	TROUGHPUT RATE		SIRO
		FCFS	LCFS	
1	15000	0.0100	0.0100	0.0100
2	55000	0.0100	0.0100	0.0100
3	22000	0.0099	0.0101	0.0100
4	78000	0.0100	0.0099	0.0100
5	34000	0.0100	0.0099	0.0100
6	61000	0.0100	0.0099	0.0100
7	92000	0.0099	0.0100	0.0099
8	27000	0.0100	0.0100	0.0099
9	100000	0.0100	0.0099	0.0099
10	84000	0.0100	0.0099	0.0100

**TABLE IV**  
**Mean and Stdv for Test Case #1**

DISCIPLINE	MEAN	STAND. DEV.
FCFS	0.00998	0.0000327
LCFS	0.00996	0.0000527
SIRC	0.00997	0.0000353

for the 0.01 level of significance (2.821), the null hypothesis is accepted for all the queueing disciplines. However as the same statistics are greater than 1.833 the null hypothesis is rejected for all the queueing disciplines if the 0.05 level of significance is to be considered.

### 3. Test Case #2

#### a. Objective

The objective is to test CPMT for simulation of multiple servers within a servergroup. The parameter to be measured is the mean number of jobs in the system.

#### b. Simulation Model

The simulation model consists of a M/M/2 queueing model with a mean interarrival rate of 20 and a mean service rate of 10.

#### c. Simulation Results

The simulation run duration was specified by the number of events to be processed and the output results are listed in table V.

**TABLE V**  
**Simulation Results of Test Case #2**

<u>RUN #</u>	<u>NUMBER EVENTS</u>	<u>MEAN NUMBER JOBS</u>
1	25000	0.526
2	43000	0.561
3	127000	0.530
4	99000	0.561
5	63000	0.530
6	85000	0.529
7	178000	0.539
8	134000	0.517
9	225000	0.515
10	165000	0.543



#### d. Analytic Results

The mean number of jobs in the system for the M/M/2 model is described by equation 5.2 where the utilization  $U$  is computed from equation 5.3. The utilization obtained from the last equation using the mean values of the model is 0.25. Substituting this value in equation 5.2 we get 0.533 as the mean number of jobs in the system.

$$E(N) = 2*U / (1 - U^2) \quad (\text{eqn 5.2})$$

$$U = \text{service mean} / (2 * \text{interarrival mean}) \quad (\text{eqn 5.3})$$

#### e. Statistical Analysis

The mean and standard deviation of the number of jobs in the system for the samples shown in Table V are :

MEAN	STANDARD DEVIATION
-----	-----
0.535	0.016

The statistic obtained from equation 5.1 using the mean and standard deviation listed above is 0.4. As this value is less than the critical values from Table II, (1.833 and 2.821) the null hypothesis is accepted for 0.05 and 0.01 levels of significance.

#### 4. Test Case #3

##### a. Objective

The objective of this experiment is to study the CPMT behavior when simulating jobs with different priorities and served by a nonpreemptive priority queueing discipline. The output to be measured is the mean time in system (response time).

#### b. Simulation Model

The model consists of a single server queueing model in which arrivals and service time occur randomly with an exponential distribution. The workload is partitioned into three classes of jobs. Each job type has a given priority, mean interarrival time and mean service time, as listed in Table VI.

Jobs are served in a nonpre-emptive priority schedule.

**TABLE VI**  
**Characteristics of each Standard Type of Job**

<u>JCB TYPE</u>	<u>PRIORITY</u>	<u>MEAN INTERARRIVAL</u>	<u>MEAN SERVICE</u>
1	1	200	50
2	2	500	125
3	3	2000	500

#### c. Simulation Results

The simulation run duration was specified by simulation time and the sample of output results is listed in Table VII.

#### d. Analytic Results

The mean response time for all jobs and for each job type for this model are taken from [Ref. 1 :p.77], and are shown in Table VIII.

**TABLE VII**  
**Simulation Results of Test Case #3**

RUN #	SIMUL TIME	MEAN TIME IN SYSTEM			
		ALL	TYPE 1	TYPE 2	TYPE 3
1	123000	505.8	291.8	611.4	2311.8
2	155000	410.5	250.4	511.8	1623.1
3	240000	438.0	265.2	574.9	1698.3
4	480000	482.6	294.4	597.9	1888.6
5	762000	499.6	296.8	628.6	2038.4
6	261000	490.1	295.1	588.8	2046.1
7	943000	397.1	243.1	529.4	1423.1
8	335000	478.9	282.3	604.3	1984.7
9	433800	452.2	272.2	565.2	1785.2
10	692000	457.3	267.0	583.7	1847.0

**TABLE VIII**  
**Analytical Solution of Test Case #3**

TYPE	RESPONSE TIME
ALL	460
TYPE 1	275
TYPE 2	575
TYPE 3	1850

**e. Statistical Analysis**

The mean and standard deviation of response time for the samples are listed in Table IX.

**TABLE IX**  
**Mean and Stdv for Test Case #3**

<u>JOB TYPE</u>	<u>MEAN</u>	<u>STANDARD DEVIATION</u>
ALL JOBS	461.2	37.1
TYPE 1	275.8	19.4
TYPE 2	579.6	36.2
TYPE 3	1864.6	250.7

The statistic values computed from equation 5.1 using the mean and standard deviation listed in Table IX are 0.1, 0.04, 0.40 and 0.18 for all jobs and for each job type. As these values are less than the critical values 1.833 and 2.821 the null hypothesis is accepted for all jobs and for each job type for 0.05 and 0.01 levels of significance.

#### 5. Test Case #4

##### a. Objective

The objective of this experiment is to test CPMT for simulation of closed network queueing models. The output parameters to be measured are the server utilizations.

##### b. Simulation Model

The simulation model consists of a closed queueing central server model which was already described in detail in Chapter 4 of this thesis.

##### c. Simulation Results

The simulation run was specified by simulation time and the output results are listed in Table X.

**TABLE X**  
**Simulation Results of Test #4**

<b>RUN #</b>	<b>SIMULATION TIME</b>	<b>CPU</b>	<b>UTILIZATIONS HARD</b>	<b>FLOPPY</b>
1	133000	0.948	0.416	0.343
2	411000	0.953	0.416	0.339
3	325000	0.966	0.403	0.347
4	127000	0.967	0.449	0.310
5	251000	0.937	0.409	0.346
6	531000	0.968	0.423	0.343
7	301000	0.939	0.443	0.343
8	210000	0.864	0.360	0.340
9	442000	0.957	0.449	0.335
10	536000	0.974	0.436	0.332

**d. Analytic Results**

The numerical solution for the model computed by the IBM software simulation package RASQ, according to LAVENBERG [Ref. 1 :p.378], estimates the following server utilizations :

CPU ..... 0.95

HARD DISK..... 0.42

FLOPPY DISK... 0.33

**e. Statistical Analysis**

The mean and standard deviations for the server utilizations obtained from the sample are listed in Table XI.

**TABLE XI**  
**Mean and Stdv for Test Case #4**

<u>SERVER</u>	<u>MEAN</u>	<u>STANDARD DEVIATION</u>
CPU	0.947	0.017
HARD DISK	0.420	0.027
FLOPPY DISK	0.337	0.011

The statistics computed from equation 5.1 using the mean and standard deviations listed in table XI are 0.57, 0 and 2.23 for CPU, HARD DISK and FLOPPY DISK. As all these values are less than the critical value for the 0.01 level of significance (2.821), the null hypothesis is accepted for all server utilizations. For the 0.05 level of significance, as the critical value (1.833) is less than the statistic found for the FLOPPY DISK utilization (2.23) and greater than the values found for CPU and HARD DISK (0.57 and 0), the null hypothesis is rejected for the first server and accepted for the last two servers.

This result is not surprising because the branching probabilities assigned to the CPMT model were rounded from the original model.

## 6. Test Case #5

### a. Objective

The objective of this test case is to estimate the CPMT performance for simulation of more complex models. To accomplish this estimation a hypothetical terminal-oriented distributed computing system was modeled. The parameters to be measured are the system throughput and mean time in system.

## b. The System

The computer system to be modeled was adapted from TRIVEDI [Ref. 3 :p.441], and it is a distributed system which primarily services three <sup>5</sup> terminals (T). The system includes four processors, a front-end (F), a communication processor (C), a DBMS processor (D) and the principal element processor (P).

A single class of jobs is processed and there is one job assigned to each terminal. The branching probabilities are described by the matrix in Figure 5.1.

	T	F	C	D	P
T	0	1	0	0	0
F	0.8	0	0.2	0	0
C	0	0.458	0	0.333	0.209
D	0	0	1	0	0
P	0	0	1	0	0

Figure 5.1 Branching probabilities Matrix

The average think time of a terminal and the average service times for the processors are assumed to be exponential distributed with the mean values as listed in Figure 5.2.

## c. Simulation Model

The model consists of a closed queueing network with five nodes. A job spends a think time at the terminal, traverses the subnetwork of processors and when it completes

---

<sup>5</sup>A small number of terminals was simulated to facilitate the analytical solution

<u>SERVER</u>	<u>AVERAGE TIME</u>
TERMINAL	15 sec.
PROCESSOR F	0.67 sec.
PROCESSOR C	1 sec.
PROCESSOR D	5 sec.
PROCESSOR P	5 sec.

**Figure 5.2 Average Think and Service Times**

has another think time. Each processor is modeled by a servergroup with a single server. The set of terminals is modeled by a servergroup with multiple servers. The CPMT model and servergroup data form are illustrated in Figures 5.3 and 5.4.

Because the smallest time is in the 0.01 sec range, see Fig 5.2 ,the time values are multiplied by 100 for routing record data input. As the routing probabilities from a given servergroup must be represented as integers and the sum must be equal to 100, the probabilities from the branching matrix shown in Figure 5.1 are rounded to meet this criterion. The job type and routing record data form for the model are illustrated in Figure 5.5.

#### d. Simulation Results

The simulation run was specified by simulation time and the output results <sup>6</sup> are listed in Table XII.

---

<sup>6</sup>Output values are divided by 100 to convert to 1 sec. time unit



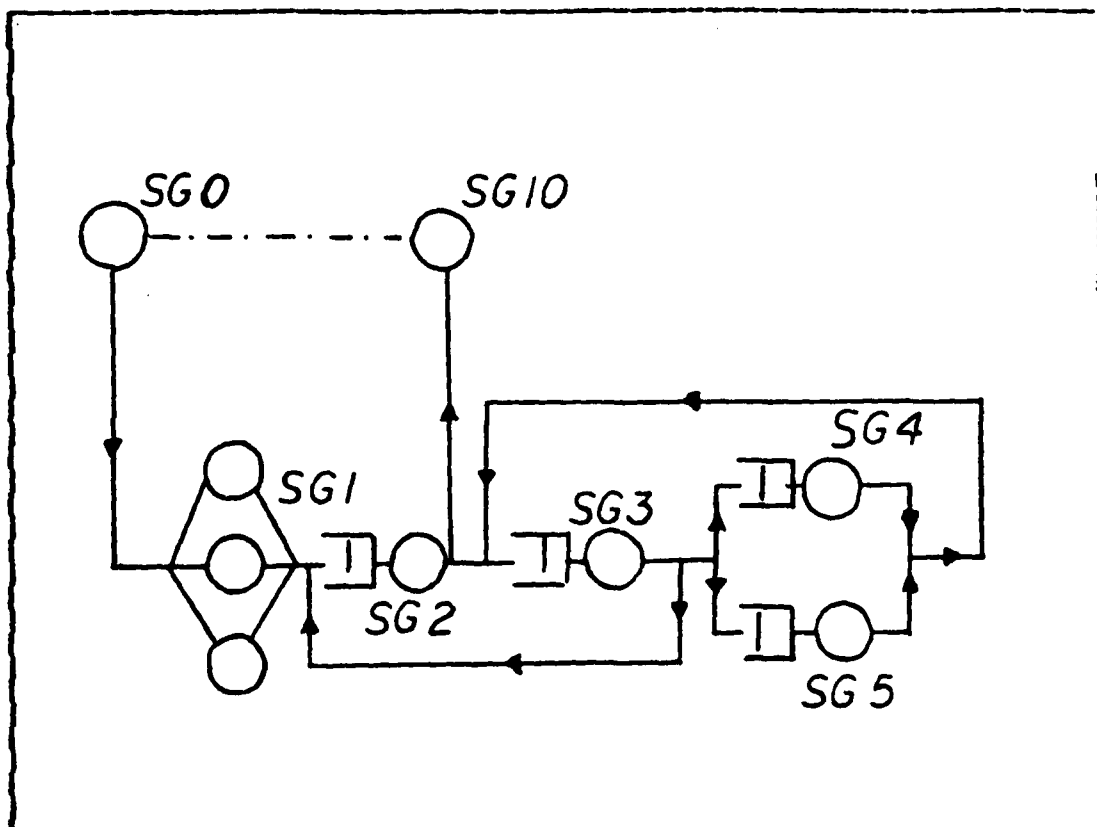


Figure 5.3 Simulation Model of Test Case #5

Simulation Number : 99	
Servergroup Number :	Number of Servers :
1	3
2	1
3	1
4	1
5	1
6	1
7	0
8	0
9	0
0	0

Figure 5.4 Servergroup Data Form for Test Case #5

```

Simulation Number : 99          Job Type Number : 1
***** JOB TYPE RECORD *****
Arrival Dist : -
Dist Param   : -
Priority      : 1
***** ROUTING RECORD *****
Servergroup : 0  1  2  3  4  5  6  7  8  9
Service Dist: -  2  2  2  2  2
Dist Param   : - 1500 67 100 500 500
Queue Disc   : -  1  1  1  1  1
Routing To   :
SG 1         100 - 80 - - -
SG 2         - 100 - 46 - -
SG 3         - - 20 - 100 100
SG 4         - - - 33 - -
SG 5         - - - 21 - -
SG 6         - - - - - -
SG 7         - - - - - -
SG 8         - - - - - -
SG 9         - - - - - -

```

Figure 5.5 Job Type and Routing Record Data Form of Test #5

#### e. Analytic Results

The analytic procedure used to solve the network model was extracted from TRIVEDI [Ref. 3].

From the branching probabilities listed in Fig 5.1 we get the following system of linear equations for computation of relative throughputs  $V_i$ 's of the network nodes :

$$V_T = V_F * 0.8$$

$$V_F = V_T + V_C * 0.458$$

$$V_C = V_F * 0.2 + V_D + V_P$$

$$V_D = V_C * 0.333$$

**TABLE XII**  
**Simulation Results of Test Case #5**

<b>RUN #</b>	<b>SIMULATION TIME</b>	<b>THROUGHPUT RATE</b>	<b>MEAN TIME IN SYSTEM</b>
1	2981300	0.16	18.707
2	1780400	0.16	18.322
3	3173500	0.16	18.211
4	2346000	0.17	17.941
5	1995000	0.17	17.763
6	2342100	0.16	18.401
7	3812000	0.17	17.973
8	8900000	0.17	17.883
9	5678800	0.16	18.199
10	2120000	0.17	18.074

$$V_p = V_c * 0.209$$

Choosing  $V_r = 1$  and solving the system of equations we find the following relative throughput for the remaining nodes :

$$V_r = 1.25$$

$$V_c = 0.546$$

$$V_p = 0.182$$

$$V_d = 0.114$$

The relative utilization of node i is given by the equation 5.4 where  $V_i$  is the relative throughput and  $E(S)$  the service time.

$$\rho_i = v_i * E(S) \quad (\text{eqn 5.4})$$

Substituting the service times from figure 5.2 and throughput in equation 5.4 we get the following relative utilizations :

$$\rho_T = 15$$

$$\rho_F = 0.83$$

$$\rho_C = 0.54$$

$$\rho_D = 0.90$$

$$\rho_P = 0.57$$

The average system throughput is given by equation 5.5 where  $M$  is the number of terminals and  $C$  is the normalization constant.

$$\text{TROUGHGHPUT} = C(M - 1)/C(M) \quad (\text{eqn 5.5})$$

The computation of normalization constants is performed by a recursive scheme based on the equations 5.6, 5.7 and 5.8, where  $c$  is the number of servers at node  $i$ , and  $B_i(k_i)$  the joint probability of  $k$  jobs at node  $i$ .

$$B_i(k_i) = \begin{cases} k_i! & k_i < c_i \\ c_i! c_i^{k_i - c_i} & k_i \geq c_i \end{cases} \quad (\text{eqn 5.6})$$

$$R_i(k) = \begin{cases} \rho_i / B_i(k) & k \neq 0 \\ 1 & k = 0 \end{cases} \quad (\text{eqn 5.7})$$

$$C_i(j) = \begin{cases} \sum_{k=0}^{R_0(j)} C_{i-1}(j-k) R_i(k) & i \neq 0 \\ R_0(j) & i = 0 \end{cases} \quad (\text{eqn 5.8})$$

The values obtained for  $C(2)$  and  $C(3)$  are 160.16 and 965.8. Substituting these values in equation 5.5 we found 0.166 as the analytic throughput rate.

The analytic response time is obtained from the equation

$$\text{RESPONSE TIME} = M / \text{TROUGHPUT} - \text{THINK TIME}$$

Substituting the number of terminals 3, throughput 0.166 and think time 15 sec. in the equation above, we get 3.116 sec. as the analytic response time. As the parameter to be compared is the mean time in the system we have to add the average think time, 15 seconds to this value to find the analytic result which is 18.116 sec.

f. Statistical Analysis

The sample mean and standard deviation for throughput and mean time in system are listed in Table XIII

TABLE XIII		
Mean and Stdv of Test Case #5		
	<u>TROUGHPUT</u>	<u>TIME IN SYSTEM</u>
MEAN	0.165	18.147
STDV	0.005	0.427

The statistics obtained from equation 5.1 using the values of Table XIII are 0.625 and 0.23 for throughput and response time. As these statistics are less than the critical values for the 0.01 and 0.05 levels of significance (2.821 and 1.833), the null hypothesis is accepted for both accuracy bounds.

## 7. Conclusions

From the results of statistical tests performed on the population mean for different output parameters and simulation models we conclude that the simulation results do not differ significantly (at 0.01 and 0.05 levels of significance) from what would be the analytic results.

The accuracy of results could be improved by extending the precision of the branching probabilities to bring them in closer correlation with the simulated systems.

The complexity of the analytic procedure which was required to obtain numerical solution for the performance parameters of the distributed sytem (Test Case #5) illustrates the advantage of using simulation techniques for evaluation of computer systems.

## VI. CONCLUSION

Maintenance of a software product is considered to begin from the time that the program becomes operational and is primarily concerned with changes to reflect expansion of requirements. This thesis was intended for enhancement of an operational simulation program (CPMT) in areas such as modeling capability, simulation run flexibility, processing efficiency and user friendliness.

A new class of queueing network models, closed network, and five additional queueing disciplines, Last Come First Serve, Serve in Random Order, Nonpreemptive Priority, Short Processing Time First, and Weighted Short Processing Time First were made available in the new version of CPMT to increase the modeling flexibility. However further enhancement could be done in this area. Extension of the network models in order to include multiple sources and/or sinks, and passive queues are examples of potential topics for enhancement. Assumptions of the simulator design such as the server be always serving a job when jobs are present and infinite capacity of the servergroups may not be true in some model applications and therefore they can also be the object of research. Finally, pre-emptive queueing disciplines such as Last Come First Served Preemptive Resumed (LCFSPR) and Preemptive Priority (PP) are not implemented and could be useful for modeling of some real systems.

The modified program provides alternative methods of specifying the simulation run duration. Simulation time and number of events to be processed are new options to define the period of time a simulation is to be run.

The memory requirements of the program were significantly reduced by changing the space complexity of

the algorithm for job generation. Sizable simulations can be run in the new version without any system limitation.

A large number of changes was introduced in the program operation in response to criticism of CPMT users and as result of our intensive utilization of the program. The evaluation of the current program user friendliness can only be done by further CPMT utilization.

The accuracy of the results was discussed in detail in the last chapter and demonstrates the CPMT ability to simulate computer systems represented by open or closed queueing network models. Further it has been demonstrated that the time and work required for computer modeling and simulation using CPMT are relatively constants regardless of the complexity of the simulation models to be run.



# LIST OF REFERENCES

1. Lavenberg, S.S., and others, Computer Performance Modeling Handbook, Academic Press, 1983
2. Pagel, K.A., Computer Performance Modeling Tool, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1984
3. Trivedi, K.S., Probability and Statistics with Reliability, Queueing and Computer Science Applications, Prentice-Hall, 1982
4. Saver, C.H., Chand K.M., Computer Systems Performance Modeling, Prentice-Hall, 1981

# INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2	
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2	
3. Chairman, Code 052 Department of Computer Science Naval Postgraduate School Monterey, California 93943-5100	2	
4. Curriculum Officer, Code 037 Computer Technology Program Naval Postgraduate School Monterey, California 93943-5100	2	
5. Professor Alan A. Ross, Code 052 Department of Computer Science Naval Postgraduate School Monterey, California 93943-5100	2	
6. Lt. Barry Frew, Code 054 Department of Administrative Science Naval Postgraduate School Monterey, California 93943-5100	1	
7. Jose Duarte Praça de Sofia 4, 4to 2780 Oeiras Portugal	2	

**END**

**FILMED**

**10-85**

**DTIC**